Proceedings of the Eleventh International Conference on Informatics

INFORMATICS 2011

Editors

Valerie Novitzká
Štefan Hudák

Organized by:

Slovak Society for Applied Cybernetics and Informatics

Faculty of Electrical Engineering and Informatics,
Technical University of Košice

November 16-18, 2011
Rožňava, Slovakia

informatics 2011

Proceedings of the Eleventh International Conference on Informatics

INFORMATICS 2011

Editors

Valerie Novitzká
Štefan Hudák

Proceedings of the Eleventh International Conference on Informatics

INFORMATICS'2011

Rožňava, Slovakia, November 16-18, 2011

Editors

Valerie Novitzká
Štefan Hudák
Technical University of Košice
Košice, Slovakia

Organized by

Slovak Society for Applied Cybernetics and Informatics

and

Faculty of Electrical Engineering and Informatics,
Technical University of Košice

Partners

Central European Journal of Computer Science
Acta Electrotechnica et Informatica

Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
Phone: +421-55-63 353 13
Fax: +421-55-602 2746
URL http://kpi.fei.tuke.sk

ISBN 978-80-89284-94-8

PREFACE

The eleventh event of the international scientific conference Informatics 2011 is an international forum for presenting the original research results, for sharing the experience and to exchange the ideas transferring from theoretical concepts into real-life domains by scientist and experts working in computer science and informatics. It also provides an opportunity for young researchers to demonstrate the achievements and to discuss their results at international scientific forum. The main topics of the conference are as follows:

- Computer Architectures
- Computer Networks
- Theoretical Informatics
- Programming Paradigms, Programming Languages
- Software Engineering
- Distributed Systems
- Computer Graphics and Virtual Reality
- Artificial Intelligence
- Knowledge Management
- Information System Research
- Applied Informatics and Simulation
- Informatization of self-government in the Information Society

All submitted papers have been reviewed by two members of the conference Program Committee. The proceedings also contain two invited lectures that were delivered by eminent experts in their respective fields. Selected papers will be reviewed by two independent international reviewers and printed in special issues of the following international scientific journals *Acta Electrotechnica et Informatica* and *Central European Journal of Computer Science*.

We would like to express our thanks to all the authors, reviewers and, above all, our invited speakers that contributed and guaranteed the high and professional level of the conference. We are very grateful to our sponsors for the financial and material support, too.

The eleventh conference event is organized by *Slovak Society for Applied Cybernetics and Informatics* and the *Faculty of Electrical Engineering and Informatics, Technical University of Košice* and it is supported and sponsored by *Association of Slovak Scientific and Technological Societies*.

The conference Informatics 2011 is held in Rožňava, the regional capital of Gemer, the region with rich and famous historical and cultural monuments.

We are confident that inside these covers you will find relevant papers in the field of your interest. We also look forward to your participation at the next events of the Informatics conference.

November 2011

On the behalf of
Program and Organizing Committees

Valerie Novitzká, Štefan Hudák

CONTENTS

Section: Knowledge Management

Section: Applied Informatics and Simulation

Section: Information System Research

From Keyword Search Towards Exploratory Information Seeking, And Beyond

Pavol Navrat
Institute of Informatics and Software Engineering
Slovak University of Technology
Bratislava, Slovakia
navrat@fiit.stuba.sk

Abstract—**This paper surveys principal concepts involved in various approaches to web search. There are many attempts to improve key word search. There is the concept of exploratory search, which represents a shift towards more complex view of the interested fellow's role, widening her options. There is proposed a more radical shift towards viewing information seeking as traveling in the digital information space involving both web and digital libraries.**

Keyword search; exploratory search; social web; semantic web; digital space; digital library; interested fellow

I. INTRODUCTION

Keyword search is a standard approach to information retrieval that has been used for a long time in a pre-web age. It is therefore not surprising that this scheme has been adopted by search engines operating on the web. It should be noted, however, that other approaches, e.g. one based on categorization, are attempted, too, but still keyword search is undoubtedly currently the dominant metaphor. After inception of the web it took some time to realize (at least some of) its potential. There was a growing realization that web could be, and gradually has been becoming, an immense well of information. Soon people were caught by surprise by the speed of growth and its apparent sustainability. It has been quite common (including myself) to write articles starting motivation with expressive statements that everything is on the web nowadays, implicitly including also the interpretation that what is not on the web is as good as nonexistent so the closed world assumption holds for the web.

Things are more complicated. If the window to see contents of the web is the standard search engine interface that involves both preprocessing of the query and then postprocessing of the results, then our view of the web is limited be capabilities of this interface and the underlying search engine. It is known that not all the contents of the web is indexed by search engines, but indexing is a precondition to be retrievable. There are vast areas of the web that are not retrievable, cf. the notions of shallow and deep web. There is also another temporal reason since indexing is always behind in time to occurrence of new content in a web page, so for some initial period of time even information in the shallow web that is retrievable under standard circumstances is not visible.

This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-0208-10 and by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0508/09. This contribution/publication is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

Besides these principal space and time limitations of ability to be retrieved there are other grounds for suboptimal performance of information search in the web, be it expressed in terms of precision, recall or some other measure. This is one of the hottest topics of research today. For general research public, improving the actual search engine is not a straightforwardly open topic of research due to proprietary nature of the most popular search engines which are mainly researched, designed and amended by their competing owners. It is therefore left to orient most of its creative talents to interface, including both preprocessing and postprocessing in the above suggested sense.

We mention some of the research endeavours in both these areas. However, despite many interesting results of research that have lead to improvements, such approaches are somehow limited. Besides the effective exclusion of search engines, we argue that the view as outlined above is too narrow. Information seeking should be viewed in a much broader sense, not as a (whatever sophistically) preprocessed and postprocessed keyword search. In recent years, the notion of exploratory search has attracted much attention as an alternative to the standard keyword search. However, we shall show that it is much more a complement and an advancement of the standard approach. Finally, we outline some possible directions beyond these approaches. In particular, we present our idea to view information seeking as traveling in digital space.

II. PREPROCESSING

In a sense, preprocessing, i.e. some kind of preparation of the input data to be subsequently processed by a search engine is more important than any manipulation that occurs afterwards. Let us bear in mind that ultimately, output is determined in a significant extent by input. If something is misplaced in the modified list of keywords, which is input to the search engine, it will determine its output for better or worse. Any improvements of output then are just attempts to amend results that are inherently limited by input imperfection. Therefore, much emphasis has been placed in research on ways how to prepare input to the search engine.

One direction of research is query expansion. Original query is expanded to increase chances that the search engine will return results that better reflect the expectation or intention of the interested fellow. What are possible sources that are

basis for expansion? And which of them are most effective to achieve improvement in information seeking? There are several approaches to investigating these issues.

One idea is to augment the query with some additional information that reflects interested fellow's intention. For example, key words such as java or apple in a query can mean very different things depending on if the interested fellow is interested in geography or programming, computers or pomology. If we want to have scope of possible interpretations of the query narrowed but do not want to be dependent on receiving any further information from the interested fellow, we may take previous searches to extract from it clues pointing at interested fellow's intentions. Query logs can be analysed so that interested fellow's intention is automatically identified [33]. Another possible source is the personal collection of text documents, emails, cached Web pages, etc. constituting her desktop or Personal Information Repository [17].

There are works favouring explicit interested fellow's feedback with the obvious weakness in dependency from the interested fellow. Luckily, interested fellow leaves often some implicit feedback [27], [29]. Of course, all such approaches attempt to contribute to personalization of search. But is personalization of search necessary? Many queries are such that almost everyone who issues them has the same thing in mind and seeks the same information. In such cases, no personalization is necessary, at least during preprocessing. There are many other kinds of queries, however, which mean different thing to different people. Even if two different interested fellows write the same query they are interested in different information. There is an ambiguity in a query. It can be analyzed and modeled. This can help identify where a query can benefit from personalization [34]. Several important user personalization approaches and techniques developed for the web search domain are presented in [24]. A simple personalization layer that improves relevancy of search was proposed in [7].

One of the approaches towards an automatic personalization of web search is proposed in [16]. They make use of PC desktop, which is source of a wealth of specific information. Having extracted it, it allows for an increased quality of user profiling. More specifically, they select personalized query expansion terms for web search using three different desktop oriented approaches: summarizing the entire desktop data, summarizing only the desktop documents relevant to each user query, and applying natural language processing techniques to extract dispersive lexical compounds from relevant desktop resources. Related issue supportive to these approaches is text mining [70][72][80][81].

A novel query expansion algorithm is proposed in [32]. It acts as preprocessing set up on the client machine. It can learn an interested fellow's preference implicitly and then generate a profile of the interested fellow automatically. When the interested fellow inputs query keywords, more personalized expansion words are generated by their algorithm. These words together with the query keywords are then submitted to a popular search engine.

A related but somewhat different approach from the interested fellow's point of view is assisted query formulation. The interested fellow is assisted right after having formulated first piece of text expressing her interest. In a series of steps: query formulation – find – re-formulation, the text (search query) is becoming finer and more accurate [35].

As a further step in the level of assistance, query suggestion is another area of research. It is most widely known for popular queries but the usual experience shows a sharp deterioration for less frequent ones. To improve rare query suggestion, other sources of information are sought to leverage implicit feedback from interested fellows in the query logs. In [36], a combination of interested fellow's click and skip (of the returned URL's) information is used as an implicit feedback.

III. CONTEXT

In approaches using personalization, there is quite often mentioned context as another important concept. It shows its meaning immediately: (*con* - with) *text*, hinting at the surrounding situation for interpretation of the text. Context can be thought of as all the circumstances, the data and information that are somehow relevant to the event or fact. For the purpose of the particular research that we overview here, this definition is too vague. A very eleborated and formalized model of context has been proposed by [37]. They propose a context model distinguishing three distinct layers: knowledge layer, state layer, contextualization layer and, in addition, they add a fourth layer describing context-based adaptation. There are following components needed for this conceptual context model:

(a) Information about the current situation/state provided by sensing components
(mapping internal and external information sources into state objects).
(b) Background information about the (application) domain.
(c) Contextualization rules to constitute a contextualized state.
(d) Adaptation rules that define a set of meaningful adaptations according to the contextualized state.

In relation to web search, Lawrence [22] describes the context of a query as for example, the education, interests, and previous experience of a user, along with information about the current request.

Several approaches have been taken to using context in web search [21][8][9]. Among them, query rewriting, iterative filtering meta-search, rank biasing seem to be dominant. Query rewriting is based on appending keywords form context to search query as a string and submitting the augmented query to standard search engine.

Iterative-filtering meta-search does not augment a query but rather generates many different sub-queries and submits them to several search engines. Upon receiving results, it re-ranks them and aggregates into one set. Rank biasing supposes that a query and context of keywords are sent to modified search engine as a complex query. Having received documents matching query it then re-ranks them by fitness to context. More generally, query preprocessing opens room for context to be a source for query augmentation. A query is seen as a short list of keywords. To rewrite it, we have to identify additional keywords.

The concept of context is quite general and, as we noted earlier, often a loosely defined one. Interested fellow's interests, past searching experience etc. can be considered part of it. There are also other views, concentrating more on query itself [15]. Still other approaches employ collaboration of interested fellows [31]. There are also investigated completely different forms of search that require specialized user interface (e.g. facet browsers [47] or one-page search engines).

In a sense, web search adjusted in one way or another to a context as determined by the interested fellow is a personalized search. The other way round, when the interested fellow possesses a personal profile represented in a processable way, e.g. in form of ontology, an ample room opens for personalized web search, too [28]. Our point is, however, that there is a room for improving results of web search even if no concepts of semantic web are employed. It is possible to expand a query in a quite complex way based on personal information [17].

IV. POSTPROCESSING

Personalization and context information are two major sources of information that can be used not only in the preprocessing phase.

After a usual search engine returns a list of results, there are several ways how to proceed. The list can be modified to better reflect personal preferences of the interested fellow. Or the list can be adjusted to the context of the search. These both can be viewed as examples of postprocessing. Besides these and possibly other ways of postprocessing, there is another important role for this phase: to gather additional information that will enable better preprocessing in the next round.

An important area of research is to investigate ways of re-ranking results. In [25], interested fellow's browsing behaviour is source for data mining frequent access patterns. In accordance with her interests mined and feedbacks acquired, they propose Personalized PageRank for dynamically adjusting the ranking scores of web pages.

There are attempts to employ semantics description languages for representation of interested fellows' profiles. In [18], there is described a personalized search approach involving a semantic graph-based interested fellow's profile issued from ontology. Interested fellow's profile refers to her interest in a specific search session. It is built using a score propagation that activates a set of semantically related concepts and maintained in the same search session using a graph-based merging scheme. Personalization is achieved by re-ranking the search results of related queries using the user profile.

A possible approach to augmenting an individual interested fellow's profile is by using data from other interested fellows. In [30], they studied whether groups of people can be used to improve relevancy, or in general quality of search. They explore the similarity of query selection, desktop information, and explicit relevance judgments across people grouped in different ways. As could be expected, some groupings provide valuable insight into what members consider relevant to queries related to the group focus. On the other hand, it can be difficult to identify valuable groups implicitly.

V. EXPLORATORY SEARCH

Conventional approaches to targeted search suggest that the interested fellow knows what she is looking for and formulates a search query by using keywords for search engine. The main problems in current approaches are limited opportunities of query construction using keywords, search query ambiguity, and the minimum support for query modification and results viewing, and the fact that interested fellows often cannot say in advance what they are looking for [39]. We unwittingly just guess those keywords that might be found on pages with information that we want to obtain, with sufficient relevance that it will be placed at top positions of the list of search results. In targeted search, search engines improve the search success especially by complementing or reformulating the queries or by reordering the results. Among more sophisticated approaches belong e.g. support of query disambiguation [40] or advanced search using clustering [41].
Problem formulation of the initial query for a search engine when the interested fellow does not know precisely to specify the search target is addressed by the exploratory search approaches [42], e.g. by search based on the views across facet browsers mSpace, RelationBrowser++, by search using the examples in the IGroup [43], by support of query modification or by advanced methods of search results browsing in VisGets [44].

VI. SOCIAL ASPECTS

Originally, a website was only a content provider. Gradually the purpose of a web site changes and a web browser becomes a gateway to a variety of platforms and information channels. Web 2.0 has given interested fellows an ability to actively participate in the creation and organization of the web. It has also become another place, where an important role is played by communities - those that are based on real-world social relations, as well as those which are purely virtual and created by ad-hoc grouping of similarly behaving interested fellows [2][4][5]. Search, which relies on the links between interested fellows can efficiently tackle the problem of query disambiguation, can reorder the search results or can add to the query special community annotations [48].
Current research in the field of search and recommendations has been refocusing more on the usage of the user interactions in order to enrich the searched information space, which allows the search refinement [64]. These methods include the use of annotations in the context of social relations [49], temporal properties of interaction with objects and information sources, explicit and implicit evaluations of objects by users [50], the automatic derivation of hierarchical and overlaying categorizations [51] and the use of highly domain-specific characteristics (e.g. geographical distance or spacial data [71]).
Semantic web is based on the concept of ontology and a lot or research has been going on, including [73][65][69]. Searching the semantic web has been investigated [78], including various forms of semantic search [45] or recommendation [11]. Social navigation is also one of the relevant issues [77]. When considering social aspects, it is also important to note the

potential of collaboration for effective information retrieval [63].

Another interesting concept emerges described loosely as collective wisdom or wisdom of crowds. One way is to use the knowledge that users "encode" in folksonomies, for example in social tagging. With sufficient number of users and marked resources they begin to form genuine relationships between brands, resources and users [52].

VII. Interested Fellow Model

In almost any approach to web search, there arises the question how to model the interested fellow and her context. Traditional approaches to interested fellow modeling (a.k.a. user modeling) are not appropriate in the case of web and digital libraries as they do not count with the large and dynamic information space. The solution is to come out from so-called open corpus approach [53] which is based on two-level model of the interested fellow (document level and conceptual level). Effective combination appears to be based on tags, keywords and more complex formal models which allow covering various dimensions of the interested fellow [54]. Many other works include e.g., [1][68].

VIII. Visualization of Search Results

Both targeted and exploratory search are directly related to the effective visualization of search results overview and then the results themselves (documents, pages, information). Current approaches in the exploratory search, in addition to traditional tabular display of search results, make a significant focus on supporting the process of searching through faceted browsers like mSpace [55], RB or OntoViews [56], on improving the user orientation and understanding of information using link or content annotations.

Essential is also the support for creation of visual search queries by using approaches based on views, visualization of search and browsing history [57] with graphs and interactive visualization of relationships between search results using (hierarchical) graphs [58]. Advanced visualization approaches involve information preprocessing using clustering, for example in the search engine clusty.com, concept analysis, or reasoning context. In the area of semantic web, Tabulator [59] addresses linking of several visualization approaches using nested charts and maps. Support is still limited to individual domains, often with a manually created metadata, and without taking the dynamics into account.

IX. And Beyond (Future work)

Finally, let us outline some possible future directions of fundamental research in area of information retrieval based on search in large information spaces. Interested fellow receives or derives information in the process of her interaction within the information environment including friends and people with similar interests. Cognitive metaphor of traveling in the digital space describes an interested (often curious) fellow who travels in the web or visits digital libraries, sometimes purposefully, sometimes even without a specific objective, in order to obtain interesting information to augment her knowledge or learn something new [13] or just have fun. The range of possible cognitive experience is not limited.

Interested fellows leave traces in the digital space, sometimes even without being aware of it. For example: evaluations, recommendations, annotations, inscriptions on a virtual wall [74][75]. Interested fellows communicate with others, forming communities of those sharing interests. They express their views, write emails, blogs or microblogs [76]. Even a track record of a journey in the digital space is important information, especially if complemented with knowledge of how successful this traveling was.

All this is potentially applicable in order to improve the cognitive structure of semantics for interested fellows as information seekers (consumers), or digital space travelers. The potential of digital libraries is waiting for the revelation particularly through their integration, stressing the importance of interested fellow's information behaviour [82][83]. The web can provide distributed space for the unification of meaning across different libraries [84]. The space, with its information seeking inhabitants forms a world, which has ecological dimensions worth studying [85]. Forward-looking "single" digital world of information resources can use advanced search features, personalization, social networking, or context consideration.

In such an environment it is important to consider methods of presentation and visualization of information, in particular of a response to the query. A very promising method seems to be multi-paradigm visualization, where depending on the presented information and the state of the process of searching, the most appropriate form according to the nature of retrieval and context (facet browsing, graphs, visualization, creation of clusters and conceptual models, specific browser content) is selected [86][87].

So far, the known search methods do not provide sufficiently precise results and results sufficiently covering the relevant subspace (cf. precision, recall). This is largely a consequence of the way how people use the search engines - what queries they submit. Moreover, the amount of information on the web increases rapidly. Therefore there is a permanent demand to improve methods for searching and accessing information. Nowadays, unstructured text is still the prevailing form of storing data in the Web. Such a method of content storing does not support automatic processing. There are several research directions that focus on accessing of the web content using various engines and tools in order to enable automatic processing of the content (the semantic web). However, the results have a limited scope for specific domains and specific applications.

Development in this area matures to the point where the web is becoming so important and at the same time still unknown phenomenon that some identify it as a separate, original object of investigation. There are even initiatives to establish web science [38] as a new scientific discipline.

We formulate a working metaphor of cognitive traveling in digital information space, covering web and digital libraries. The concept of digital (information) space is a useful

abstraction, because any activity in space is usually associated with some movement and moving in space can be seen as traveling. Even those interested fellows, who know what they are looking for, can find themselves in a situation that before they find what they originally have been seeking, they encounter an interesting link or information and unconsciously start searching in a new direction albeit with less accurate search target. However, there are also such interested fellows who do not know exactly what they want, but they like wandering in the information space, leaving navigation to immediate evaluation of interestingness of what they see or read.

Obviously, the outlined view of traveling in digital information space is just one of several views presented or emenging. A very comprehensive view of interactive information retrieval is presented in [6]. An interesting view of information-seeking funnel is proposed in [10]. Undoubtedly, there is much need for future work in this area.

ACKNOWLEDGMENT

I wish to express my gratitude to my colleagues, especially to Michal Barla, Maria Bielikova, Ladislav Hluchy, Michal Laclavik, Jan Paralic, Jela Steinerova, Jaroslav Susol and Michal Tvarozek, who contributed to the unpublished project proposal [60], parts of which are included in this paper.

REFERENCES

[1] A. Andrejko, Novel Approaches to Acquisition and Maintenance of User Model. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 1, No. 1 (20009), 1-10.

[2] M. Barla, Towards Social-based User Modeling and Personalization. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, No. 1 (2011) 52-60

[3] A. Broder, A taxonomy of web search. SIGIR Forum 36, 2 (September 2002), 3-10.

[4] E. H. Chi: Information Seeking can be Social. In: National Science Foundation workshop on Information-Seeking Support Systems (ISSS), June 26-27, 2008, Chapel Hill, NC, 39-45.

[5] B.H. Evans, E.H. Chi, An elaborated model of social search. Information Processing and Management, (2009), article in press.

[6] P. Ingwersen, The User in Interactive Information Retrieval Evaluation. In: M. Melucci, R. Baeza-Yates (eds.), *Advanced Topics in Information Retrieval,* The Information Retrieval Series 33, Springer-Verlag Berlin Heidelberg 2011, 83-107.

[7] M. Kajaba, P. Navrat, D. Chuda, A Simple Personalization Layer Improving Relevancy of Web Search. Computing anf Information Systems Journal. Vol.13 (2009), No. 3. 29-35.

[8] P. Navrat, T. Taraba, Context Search. In: Y. Li, V.V. Raghavan, A. Broder, H. Ho: 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology: IEEE Computer Society, 2007. 99-102.

[9] Navrat, P., Taraba, T., Bou Ezzeddine, A., Chuda, D.: Context Search Enhanced by Readability Index. In: IFIP Series. ISSN 1571-5736. Vol. 276: Artificial Intelligence in Theory and Practice II. (2008). Springer Science+Business Media, LLC. 373-382.

[10] D.E. Rose, The Information-Seeking Funnel. In: National Science Foundation workshop on Information-Seeking Support Systems (ISSS), June 26-27, 2008, Chapel Hill, NC, 32-36.

[11] J. Suchal, P. Navrat, Full Text Search Engine as Scalable k-Nearest Neighbor Recommendation System. In: Artificial Intelligence in Theory and Practice III IFIP Advances in Information and Communication Technology, 2010, Volume 331/2010, 165-173.

[12] Y.L. Theng, H. Thimbleby, M. Jones, M., "'Lost in hyperspace': Psychological problem or bad design?", APCHI'96, 387-396, Singapore, 1996.

[13] J. Tvarozek, Bootstrapping a Socially Intelligent Tutoring Strategy. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, No. 1 (2011) 33-41.

[14] M. Tvarozek, Exploratory Search in the Adaptive Social SemanticWeb. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, No. 1 (2011) 42-51.

[15] J. Bai, J. Nie. (2008). Adapting Information Retrieval To Query Contexts. *Information Processing Management*, 44(6): 1901-1922.

[16] P. Chirita, C.S. Firan, W. Nejdl. (2006). Summarizing local context to personalize global web search. In *Proceedings of the 15th ACM international Conference on information and Knowledge Management CIKM '06*. ACM, New York: 287-296.

[17] P. Chirita, C.S. Firan, W. and Nejdl, W. (2007). Personalized Query Expansion For The Web. In Proceedings of the 30th Annual international ACM SIGIR Conference on Research and Development in information Retrieval SIGIR '07. ACM, New York: 7-14.

[18] M. Daoud, L. Tamine-Lechani, M. Boughanem, B. Chebaro. (2009). *A session based personalized search using an ontological user profile.* In Proceedings of the 2009 ACM Symposium on Applied Computing SAC '09. ACM, New York: 1732-1736.

[19] F. Kawsar, K. Fujinami, S. Pirttikangas, T. Nakajima. (2006). Personalization and Context Aware Services: A Middleware Perspective. In *Proc. 2nd International Workshop on Personalized Context Modeling and Management for Ubicomp Applications (UbiPCMM) in conjunction with Ubicomp 2006*. Orange County, California.

[20] G. Koutrika, Y. Ioannidis. (2005). A Unified User Profile Framework for Query Disambiguation and Personalization. In: P. Brusilovsky, C. Callaway, A. Nurnberger (Eds.): *Proc. Workshop on New Technologies for Personalized Information Access, part of the 10th Int. Conf. on User Modeling (UM'05),* Edinburgh: 44-53.

[21] R. Kraft, C.C. Chang, F. Maghoul, R. Kumar. (2006). Searching With Context. In *Proceedings of the 15th international Conference on World Wide Web WWW '06*. ACM, New York: 477-486.

[22] S. Lawrence. (2000). Context in Web Search, *IEEE Data Engineering Bulletin*, 23(3): 25–32.

[23] J. Luxenburger, S. Elbassuoni, G. Weikum. (2008). Matching Task Profiles And User Needs In *Personalized Web Search. In Proceeding of the 17th ACM Conference on information and Knowledge Management CIKM '08*. ACM, New York: 689-698.

[24] A. Micarelli, F. Gasparetti, F. Sciarrone, S. Gauch. (2007). Personalized Search on theWorld Wide Web. In P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.): *The Adaptive Web*, LNCS 4321, Springer, Berlin: 195–230.

[25] W. Peng, Y. Lin. (2006). Ranking Web Search Results from Personalized Perspective. In Proceedings of the the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services CEC-EEE. IEEE Computer Society, Washington, DC: 12.

[26] A. Salamanca, E. León. (2008). An Integrated Architecture for Personalized Query Expansion in Web Search. In *Proc. 6th AAAI Workshop on Intelligent Techniques for Web Personalization & Recommender Systems*. Chicago: 20-28.

[27] X. Shen, B. Tan, C. Zhai: (2005). Implicit User Modeling for Personalized Search. In: O. Herzog, H.-J. Schek, N. Fuhr, A. Chowdhury, W. Teiken (Eds.): *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*, ACM New York: 824-831.

[28] A. Sieg, B. Mobasher, R. Burke. (2007). Web Search Personalization With Ontological User Profiles. In Proceedings of the Sixteenth ACM Conference on Conference on information and Knowledge Management CIKM '07. ACM, New York: 525-534.

[29] J. Teevan, S.T. Dumais, E. Horvitz. (2005). Personalizing Search Via Automated Analysis Of Interests And Activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference On Research And*

Development In Information Retrieval SIGIR '05. ACM, New York: 449-456.

[30] J. Teevan, M.R. Morris, S. Bush. (2009). Discovering and using groups to improve personalized search. In *Proceedings of the Second ACM international Conference on Web Search and Data Mining,* R. Baeza-Yates, P. Boldi, B. Ribeiro-Neto, and B. B. Cambazoglu, Eds. WSDM '09. ACM, New York: 15-24.

[31] M. Tvarozek, M. Bielikova. (2008). Collaborative Multi-Paradigm Exploratory Search. In: *Proceedings of the Hypertext 2008 Workshop on Collaboration and Collective Intelligence, WebScience´08*, ACM Press, New York: 29-33.

[32] Z. Zhu, J. Xu, X. Ren, Y. Tian, L. Li. (2007). Query Expansion Based on a Personalized Web Search Model. In *Proceedings of the Third international Conference on Semantics, Knowledge and Grid SKG*. IEEE Computer Society, Washington, DC, 128-133.

[33] R. Baeza-Yates, L. Calderon-Benavides, C. Gonzales-Caro, The Intention Behind Web Queries, In: F. crestani, P. Ferragina, M. Sanderson (Eds.): SPIRE 2006, LNCS 4209, Springer 2006, 98-108.

[34] J. Teevan, S.T. Dumais, D.J. Liebling, To Personalize or Not to Personalize: Modeling Queries with Variation in User Intent, In *Proceedings of the ACM SIGIR Conference On Research And Development In Information Retrieval* SIGIR '08. ACM, Singapore: 163-170.

[35] H. Dreher, R. Williams, Assisted Query Formulation Using Normalised Word Vector and Dynamic Ontological Filtering, In: H. Legind Larsen et at. (Eds.): FQAS 2006, LNAI 4027, Springer 2006, 282-294.

[36] Y. Song, L. He, Optimal Rare Query Suggestion With Implicit User Feedback, In: WWW 2010, ACM, 901 – 910.

[37] J.M. Haake, T. Hussein, B. Joop, S. Lukosch, D. Veiel, J. Ziegler, Modeling and Exploiting Context for Adaptive Collaboration. Int J Cooperative Inf Syst, 19(1-2), 2010, 71-120.

[38] J. Hendler, N. Shadbolt, W. Hall, T. Berners-Lee, D. Weitzner. 2008. Web science: an interdisciplinary approach to understanding the web. *Commun. ACM* 51, 7 (Jul. 2008), 60-69.

[39] B.J. Jansen, A. Spink, J. Bateman, T. Saracevic. 1998. Real life information retrieval: a study of user queries on the Web. ACM SIGIR Forum , 32 (1), 5-17.

[40] G. Bordogna, A. Campi, S. Ronchi, G. Psaila. 2009. Query disambiguation based on novelty and similarity user's feedback, in WI-IAT '09: Proc. of the 2009 IEEE/WIC/ACM WI-IAT, IEEE CS, 2009, 125–128.

[41] P. Braak, N. Abdullah, Y. Xu. 2009. Improving the performance of collaborative filtering recommender systems through user profile clustering, in WI-IAT: IEEE CS, 2009, 147–150.

[42] G. Marchionini. 2006. Exploratory search: from finding to understanding. Communications of the ACM, 49 (4), 41-46.

[43] S. Wang, F. Jing, J. He, Q. Du, L. Zhang. 2007. IGroup: presenting web image search results in semantic clusters. Proc. of the SIGCHI Conf. on Human Factors in Comp. Sys. 587-596. USA: ACM Press, New York, NY, USA.

[44] M. Dörk, S. Carpendale, C. Collins, C. Williamson. 2008. VisGets: Coordinated visualizations for web-based information exploration and discovery. IEEE Trans. on Visualization and Computer Graphics, 14(6) 1205–1212.

[45] R. Guha, R. McCool, E. Miller. 2003. Semantic Search. Proc. of the 12th Int. Conf. on World Wide Web, 700-709. ACM Press, New York, NY, USA.

[46] C. Ono, M. Kurokawa, Y. Motomura, H. Asoh. 2007. A. Context-aware movie preference model using a Bayesian network for recommendation and promotion. In: UM 2007. LNCS, vol. 4511, Springer, 247–257.

[47] B. Kules, R. Capra, M. Banta, T. Sierra. 2009. What do exploratory searchers look at in a faceted search interface?. In Proc. of the 9th ACM/IEEE-CS JCDL '09. ACM, New York, 313-322.

[48] S. Barry et al. 2009. Google Shared. A Case-Study in Social Search. UMAP 2009: 283-294

[49] S. Bao, et al. 2007. Optimizing Web Search Using Social Annotations. In. WWW 2007, May 8–12, 2007, Banff, Alberta, Canada, 501-510

[50] T. Joachims, F. Radlinski. 2007. Search Engines that Learn from Implicit Feedback, IEEE Computer, Vol. 40, No. 8, August, 2007, 34-40

[51] E. Schwarzkopf. et al. 2007. Mining the Structure of Tag Spaces for User Modeling (2007). In: Data Mining for User Modeling, Workshop held at UM 2007, 63–75.

[52] P. Mika. 2007. Ontologies are us: A unified model of social networks and semantics. J. Web Sem. 5(1): 5-15

[53] P. Brusilovsky, N. Henze. 2007. Open Corpus Adaptive Educational Hypermedia. In The Adaptive Web, LNCS 4321, Springer, 671-696

[54] D. Heckmann et al. 2007. The User Model and Context Ontology GUMO revisited for future Web 2.0 Extensions, Contexts and Ontologies: Representation and Reasoning, 37-46.

[55] M.L. Wilson, M.C. Schraefel, R.W. White. 2009. Evaluating advanced search interfaces using established information-seeking models. In: J. of the American Society for Inf. Science and Technology, 60 (7), 1407-1422.

[56] E. Mäkelä, E. Hyvönen, S. Saarela, K. Viljanen. 2004. OntoViews - A Tool for Creating Semantic Web Portals. In ISWC 2004: Proc. of the 3rd Int. Semantic Web Conference. LNCS 3298, Springer, 797-811.

[57] M. Mayer. Web history tools and revisitation support: A survey of existing approaches and directions, Foundations and Trends in HCI, vol. 2, no. 3, 173–278, 2009.

[58] H.-J. Schulz, H. Schumann. 2006. Visualizing Graphs - A Generalized View. In IV 2006: 10th Int. Conf. on Information Visualization. 166-173, IEEE CS.

[59] T. Berners-Lee et al. 2006. Tabulator: Exploring and analyzing linked data on the semantic web, In Proc. of Int. Sem. Web User Interaction Workshop.

[60] P. Navrat et al., Cognitive traveling in digital space of the Web and digital libraries supported by personalized services and social networks. Unpublished project proposal, Slovak University of Technology, Bratislava 2010.

[61] A. Andrejko, M. Bieliková. Comparing Instances Of Ontological Concepts For Personalized Recommendation. In Computing and Informatics, Vol. 28, No. 4 (2009), 429-452.

[62] F. Babič, J. Paralič, M. Raček, J. Wagner. Analyzes Of Interactions And Context Of Performed Actions In Intelligent Virtual Env. Ambient Intelligence Perspectives, Vol. 2, No. 1 (2010), IOS Press, 137-144.

[63] F. Babič, J. Paralič, K. Furdík, P. Bednár, J. Wagner. Use Of Semantic Principles In A Collaborative System In Order To Support Effective Information Retrieval. LNAI 5796, Springer, 2009, 365-376, Computational Collective Intelligence, Vol. 5796.

[64] M. Barla, M. Tvarožek, M. Bieliková. Rule-Based User Characteristics Acquisition From Logs With Semantics For Personalized Web Systems. In: Computing And Informatics. Vol. 28, No. 4 (2009), 399-427.

[65] P. Bartalos, M. Bieliková. Fast And Scalable Semantic Web Service Composition Approach Considering Complex Pre/Postconditions. In: Services 2009: 2009 IEEE Congress On Services, Los Angeles, Ca, IEEE Computer Society, 2009. 414-421.

[66] Bielikova, M., Navrat, P.: Adaptive Web-Based Portal For Effective Learning Programming. In: Communication & Cognition. Vol. 42, No. 1/2 (2009), 75-88.

[67] M. Bieliková, M. Divéky, P. Jurnečka, R. Kajan, L. Omelina. Automatic Generation Of Adaptive, Educational And Multimedia Computer Games. In: Signal, Image And Video Processing. Vol. 2, No. 4 (2008). Springer London, 371-384.

[68] M. Bieliková, P. Nagy. Considering Human Memory Aspects For Adaptation And Its Realization In Aha!. In Lecture Notes In Computer Science. Vol. 4227 Innovative Approaches For Learning And Knowledge Sharing, 1st European Conf. On Technology Enhanced Learning, 2006.

[69] P. Butka. Use Of FCA In The Ontology Extraction Step For The Improvement Of The Semantic Information Retrieval. In Proc. Of Sofsem 2006: Theory And Practice Of Computer Science, Proceedings Volume II, Měřín, Czech Republic, 2006, 74-82.

[70] P. Butka, M. Sarnovský, P. Bednár. One Approach To Combination of FCA-Based Local Conceptual Models For Text Analysis - Grid-Based

Approach. In Proc. Of IEEE Int. Symp. On Applied Machine Intelligence. 2008, 131-135.

[71] S. Dlugolinsky, M. Laclavik, L. Hluchy. Towards a Search System for the Web Exploiting Spatial Data of a Web Document. In Proceedings of the 2010 Workshops on Database and Expert Systems Applications (DEXA '10). IEEE Computer Society, Washington, DC, USA, 27-31.

[72] K. Furdík, J. Paralič, F. Babič, P. Butka, P. Bednár. Design And Evaluation Of A Web System Supporting Various Text Mining Tasks For The Purposes Of Education And Research. In: Acta Electrotechnica et Informatica, Vol. 10, No. 1 (2010), 51-58.

[73] Z. Halanová, P. Návrat, V. Rozinajová. A Tool For Searching The Semantic Web For Supplies Matching Demands. In: Communication And Cognition Artificial Intelligence. Vol. 23, No. 1-4 : E-Learning III And The Knowledge Society (2006), 77-82.

[74] M. Laclavík, M. Šeleng, M. Ciglan, L. Hluchý. Ontea: Platform For Pattern Based Automated Semantic Annotation; In Computing and Informatics, Vol. 28, 2009, 555–579.

[75] M. Laclavík, M. Šeleng, L. Hluchý. Towards Large Scale Semantic Annotation Built On Mapreduce Architecture; In M. Bubak et al. (Eds.): Proc. ICCS 2008, Part III, LNCS 5103, 331-338, 2008

[76] M. Laclavík, S. Dlugolinský, M. Kvassay, L. Hluchý. Use Of Email Social Networks For Enterprise Benefit; In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), IEEE Computer Society, Washington DC 2010, 67-70.

[77] K. Matušíková, M., Bieliková. Social Navigation For Semantic Web Applications Using Space Maps. In Computing and Informatics, Vol. 26, No. 3 (2007), P. 281-299.

[78] P. Navrat, M. Bielikova, V. Rozinajova. Acquiring, Organising And Presenting Information And Knowledge From The Web. In: Communication & Cognition. Vol. 40, Nr.1-2 (2007), P. 37-44.

[79] J. Paralič, F. Babič, J. Wagner, E. Simonenko, N. Spyratos, T. Sukibuchi. Analyses Of Knowledge Creation Processes Based On Different Types Of Monitored Data. Foundations Of Intelligent Systems, LNCS 5722, Springer, 2009, 321-330.

[80] J. Paralič, M. Paralič. Some Approaches To Text Mining And Their Potential For Semantic Web Applications. In Journal Of Information And Organizational Sciences, Vol. 31, No. 1 (2007), 157-170.

[81] M. Sarnovský, P. Butka, J. Paralič. Grid-Based Support For Different Text Mining Tasks. In: Acta Polytechnica Hungarica, Vol. 6, No. 4 (2009), 5-27.

[82] J. Steinerová, J. Šušol. Library Users In Human Information Behaviour. In Online Information Review. Vol. 29, No. 2 (2005), 139-156.

[83] J. Steinerová, J. Šušol. Users' Information Behavior - A Gender Perspective. Information Research. Vol. 12, No. 3 (2007), 1-16.

[84] J. Steinerová. Relevance Assessment For Digital Libraries. Mousaion. Vol. 25, No.2 (2007), 37-57.

[85] J. Steinerova. Ecological dimensions of information literacy. *Information Research*. Vol. 15, No. 4. December 2010 Paper CoLIS 719. [Available at http://InformationR.net/ir/15-4/colis719.html]

[86] M. Tvarožek, M. Bieliková. Personalized Faceted Browsing For Digital Libraries. In: Lecture Notes In Comp. Sci. Vol. 4675 Research And Adv. Tech. For Digital Libraries, Springer (2007) 485-488.

[87] M. Tvarožek, M. Bieliková. Personalized Faceted Navigation In The Semantic Web. In: Lecture Notes In Computer Science. Vol. 4607 Web Engineering, Springer, 2006. 511-515.

Mining Moving Object Data

Jaroslav Zendulka
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
Email: zendulka@fit.vutbr.cz

***Abstract*—Currently there is a lot of devices that provide information about objects together with location-based services accumulate huge volume of moving object data, including trajectories. This paper deals with two useful analysis tasks - mining moving object data patterns and trajectory outlier detection. We also present our experience with the TOP-EYE trajectory outlier detection algorithm when we applied it on two real-world data sets.**

I. Introduction

With recent advances in positioning, telemetry and telecommunication technologies together with wide availability of devices that produce information about the position of an object in some time enormous amounts of data about moving objects are being collected and employed by many applications. Examples of such devices include mobile phones, devices with embedded GPS or sensor networks. In general, the moving object data have a form of spatio-temporal data, which was one of complex data types that gained attention of researchers after data mining techniques for relational data had proved their usefulness. Because trajectories are very important characteristics of the behavior of moving objects and currently large amounts of trajectories are collected and stored in databases, trajectory mining has become one of major challenges in data mining.

Several useful trajectory data analysis tasks have been introduced and algorithms to solve them developed. This paper deals with two of them, namely moving object data patterns and trajectory outlier detection. The objective of the former is to find moving clusters with specific properties, the objective of the latter is to detect suspicious or anomalous moving objects. We also present our experience with TOP-EYE algorithm, which is one of trajectory outlier detection algorithms. More complete overview on mining moving data was presented by Han on DASFAA 2010 [3], [4].

II. Mining moving object data patterns

Moving object pattern analysis is a data mining task the objective of which is to discover potentially useful patterns in moving object data. Such patterns can be beneficial for economic and social studies related to social and economic behaviors of people or to climate and ecological studies related to movements of animals and changes of natural phenomena. The moving object patterns can be categorized to the following categories [7]:

- *Repetitive pattern*. It is a pattern of periodic behaviors. It may be typical for some animals to have some repetitive movement patterns. But it might be difficult for biologists to discover them although currently a lot of animal movement data is currently available.
- *Relationship pattern*. This type of patterns is focused on relationships among moving individuals. The fundamental task of this type is to find groups of objects that move together. But in some cases there are also some other relationships among objects in the group.
- *Frequent trajectory pattern*. The objective of this task is to find general moving trends of all objects in a data set in terms of space and time (where, when, with what speed etc.)

Here, we will discuss in more details only relationship patterns. If we want to find groups of objects that move together, we want to find clusters of objects in space and time with specific behavior, for example that the clusters contain the same objects. The fundamental concept here is a *moving object cluster*. It can be defined in both spatial and temporal dimensions [8]:

1) a group of moving objects should be geometrically close to each other
2) they should be together for at least some minimum time duration.

These two properties can be specified by thresholds: max_r - the radius of the cluster and/or min_o - the minimum number of moving objects in the cluster; and min_t - the minimum number of consecutive timestamp snapshots in which the group of moving objects is in the same cluster. Then a moving cluster can be defined as *Moving_cluster*(min_o, min_t). There are several specific moving objects patterns referred to as relative motion patterns [4]:

- *Flock*(min_o, max_r) - at least min_o objects are within a circular region of radius max_r and they move in the same direction. It can be extended to include also a temporal constraint to *Flock*(min_o, max_r, min_t).
- *Leadership*(min_o, max_r, min_t) - at least min_o objects are within a circular region of radius max_r, they move in the same direction, and at least one of the objects was already heading in this direction for at least $min_t time$ steps.
- *Convergence*(min_o, max_r) - at least min_o objects will pass through the same circular region of radius max_r

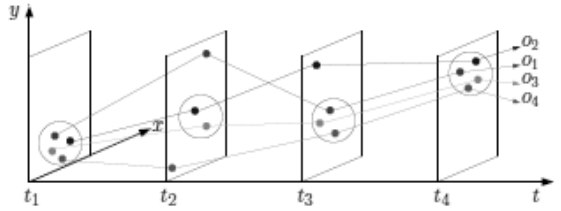


Fig. 1. The loss of interesting moving objects clusters. Adapted from [8]

(assuming they keep their direction).

- *Encounter*(min_o, max_r) - at least min_o objects will be simultaneously inside the same circular region of radius max_r (assuming they keep their speed and direction).

Disadvantage of the flock pattern is its rigid constraint in a form of the circle radium. It can result in the loss of some objects that move together with the cluster and are close to it but out of the circle defined by the radius. To avoid this was the motivation for the *convoy* pattern. It uses density-based clustering at each timestamp.

There is still another rigid constraint for both flock and convoy patterns, now in the temporal dimension, that can result in the loss interesting objects. It is illustrated in Figure 1. For example, if $min_t = 3$, no moving cluster will be found. But we would say that these four objects move together even though some objects temporarily move out of the cluster for several snapshots. To avoid such loss, the concept of *swarm* is introduced in [8]. Swarm is a group of moving objects containing at least min_o objects which are in the same cluster for at least min_t timestamp snapshots, not necessarily consecutive. For example, if $min_o = 2$ and $min_t = 3$, we can find swarm $(\{o_1, o_3, o_4\}, \{t_1, t_3, t_4\})$ in Figure 1.

An example of a tool that integrates several moving object data mining functions including periodic and swarm pattern mining is MoveMine [7].

III. Moving object trajectory outlier detection

Automatic detection of suspicious movement of objects is usually focused on detection of outliers in moving objects trajectories. Here the outlier is a trajectory which differs substantially from or is inconsistent with the remaining set of trajectories.

The objective of the trajectory outlier detection as a descriptive task is to find outliers in a trajectory database. The goal of the trajectory outlier detection as a predictive task is to decide if a trajectory of a moving object is outlier or not. In the process of detecting trajectory outliers, either whole trajectories or only their parts (partial trajectory outliers) can be considered. Both unsupervised and supervised learning can be employed to solve these tasks.

The key point of the outlier detection is how to measure the abnormality of a trajectory. In case of supervised learning, other two questions arise - how to encode a trajectory and what classifier to use. Moreover, if the outliers should be detected in real-time, it may be useful or necessary to combine various aspects of abnormality of moving objects into an unified evolving abnormality score which has the ability to simultaneously capture the evolving nature of many abnormal moving trajectories and/or detect the outlier (maybe potential) as soon as possible.

In this section, we will briefly present two approaches to measure abnormality - distance-based and motif-based. After that we will present our results with the TOP-EYE algorithm, which is able to detect top-k evolving trajectory outliers.

A. Distance-based approach

Distance-based measures are well-known in clustering where they are used to quantify the similarity of objects. Therefore they can also be used to measure abnormality of outliers. In fact, outliers can be discovered as a side-effect of clustering. They are objects out of clusters. But the main objective of clustering is to find clusters in a dataset not to detect outliers. Knorr et al. [10] introduce the concept of a distance-based outlier (*DB outlier*) and present several algorithms for mining them in a k-dimensional dataset. They define a $DB(p; D)$ outlier in a dataset T as an object O of T such that at least the fraction p of the objects in T lies greater than distance D from O. One of the real-life case studies they present in their paper is a trajectory outlier detection in a dataset extracted from surveillance videos. They consider whole trajectories. The trajectory is usually recorded and considered as a sequence of timestamped 2D points in such applications. This representation, however, may be too detailed for distance computation. Therefore, they use only several summary characteristics of the trajectory, namely start and end points locations, heading (the average, minimum and maximum values of the directional vector of the tangent of the trajectory at each point), velocity (average, minimum, and maximum velocity of the person during the trajectory). The distance of two trajectories represented as a point in this 4-dimensional space was computed as a weighted sum of differences along the dimensions. The weights were determined by domain experts.

The disadvantage of techniques based on comparing trajectories as a whole is that they are usually not able to detect outlying portions of the trajectories as it is illustrated on Figure 2. To solve this problem, in [5], the authors introduce a partition-and-detect framework and present an outlier detection algorithm TRAOD based on it. The idea is to partition a trajectory into a set of trajectory partitions referred to as *t-partitions*, and then, to detect outlying t-partitions see Figure 3.

The t-partition of a trajectory A is a line segment $L_{ij} = p_i p_j$ $(i > j)$, where p_i and p_j are two points of A. It allows a coarse granularity partitioning of a trajectory because points p_i and p_j are not necessarily two consecutive registered points of the trajectory. The authors present a two-level trajectory partitioning strategy to speed up outlier detection. First, a coarse granularity partitioning is applied. It is based on a principle referred to as a MDL principle. It leads to a partitioning which is a good trade-off between preciseness and conciseness. The

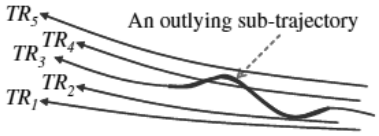


Fig. 2. An example of an outlying sub-trajectory. Adapted from [5]

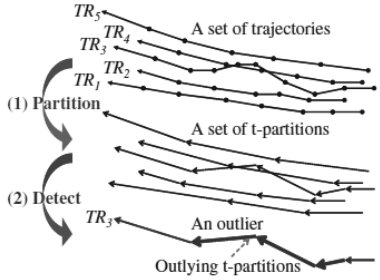


Fig. 3. The concept of a partition-and-detect framework from [5]

course granularity partitioning allows early pruning of many portions of trajectories. Only t-partitions that are likely to be outlying are partitioned into fine t-partitions and inspected.

A trajectory A is said to be a close trajectory to a t-partition L_j if the total length of t-partitions of A which are in a distance less than a threshold D from L_j is greater or equal than the length of l_j. The definition of an outlying t-partition is similar to the definition of $DB(p; D)$ outlier mentioned above. A t-partition L_i is outlying if there is a fraction p of trajectories in the database which are not close to L_i. The parameter p is given by a user. A transaction outlier is then defined as a trajectory the fraction of outlying t-partitions of which is greater or equal that a user specified threshold.

The authors also extend their definition of closeness by incorporating the density of trajectories in order not to favor t-partitions in dense regions over those in sparse ones. Therefore, this technique can be classified as hybrid, not pure distance-based. In addition, they introduce a distance function composed of three components: the perpendicular distance, the parallel distance and the angel distance. It makes possible to consider two types of outliers: positional outlier and angular outlier. They differ in the weights of the components of the distance function applied in detection.

The experimental evaluation of TRAOD on two datasets containing hurricane track data and animal movement data proved promising results and good performance characteristics. We will mention some of these results in Section III-C where we present our results obtained applying an algorithm TOP-EYE.

Another interesting approach where distance measure is derived from the idea of Minimum Hausdoff Distance is published in [9]. This distance function considers the direction and velocity of objects. Moreover, R-Tree is used to reduce the costs of its computation.

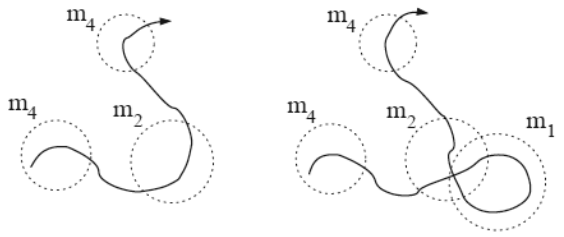


Fig. 4. An example of two trajectories and motifs they express. Adapted from [6].

B. Motif-based approach

The motif/based approach is represented by a ROAM (*R*ule- and M*o*tif-based *A*nomaly Detection in *M*oving Objects) [6]. Trajectories are expressed using discrete pattern fragments called motifs here. A rule-based classifier which accepts features derived from sequences of *motifs* with additional attributes is then used to classify trajectory outliers.

The concept of a motif is illustrated in Figure 4. The two trajectories in the figure have similar shapes except that the right one has an extra loop, The trajectories could be partitioned into such fragments that the similarity could be detected. Provided that there is a pre-defined set of representative fragments - motifs, the trajectories can be represented using the motifs. In our example both trajectories consist of the same motifs m_2 and m_4, the right one contains another motif m_1.

The framework ROAM includes three modules providing functionality for three basic steps of the trajectory anomaly detection process: motif extraction, mapping to features and classification.

First, motifs of the input trajectory data set must be extracted and trajectories mapped into a corresponding motif space. ROAM uses a sliding window to partition the trajectories in the data set. After that, clustering is used to find representative sets - motifs. Once the motifs of a given input data set are identified, the transactions are compared with the set of motifs using a sliding window of the same size as for motifs identification. The Euclidean distance is used to measure similarity. Let w be a fragment of a trajectory A in the sliding window and m a motif. We say that the motif m is expressed in the trajectory A if $\|w - m\| \leq \epsilon$, where ϵ is a parameter specified by a user.

In ROAM a trajectory is represented by a sequence of so called mofif expressions. Each motif expression has the form of $(m_i, t_{start}, t_{end}, l_{start}, l_{end})$, where m_i is the motif identification, t_{start} and t_{end} are starting and ending times, and l_{start} and l_{end} are starting and ending locations. The complete representation of the trajectory is referred here to as the *motif trajectory*. Moreover, for each motif expression, a set of other attributes that may be useful for classification can be specified.

Once the motif expressions have been extracted, motif trajectories must be mapped to a feature space that will be the input of a classifier. Because the set of different pairs of

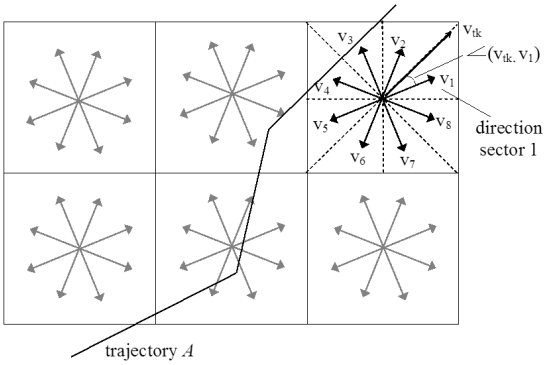


Fig. 5. Direction distance measure in TOP-EYE

attribute - attribute_value, which is the base of the mapping, can be large, the authors propose feature generalization to reduce dimensionality.

Finally, the classifier is used to the anomaly detection. The authors proposed a rule-based classifier CHIP (*C*lassification using *HI*erarchical *P*rediction Rules).

C. Evolving trajectory outlier detection

The objective of the evolving trajectory outlier detection approach is to identify evolving outliers at very early stage with relatively low false positive rate. The concept of evolving trajectory outlier was introduced in [2] where the authors present a top-k evolving trajectory outlier detection method named TOP-EYE. The method continuously computes the outlying score of a tested trajectory with respect to other trajectories in the database. The outlying score can be defined based on moving direction or density of trajectories. Here we focus on direction-based outliers.

To compute the score effectively, the continuous space of the monitored area is discretized. Moreover, a probabilistic model is used to represent directions of trajectories in the database in each cell of the discretized space grid. Each cell is partitioned into eight direction sectors, each with an angle of $\pi/4$ (see Figure 5). The goal is to represent the direction information in a cell *c* by a vector:

$$\mathbf{g} = (p_{1c},\ p_{2c},\ p_{3c},\ p_{4c},\ p_{5c},\ p_{6c},\ p_{7c},\ p_{8c}), \qquad (1)$$

where $p_{ic} \quad (i = 1, ...8)$ is the frequency of moving objects trajectory of which has direction from sector *i* in cell *c*. The frequencies are computed for trajectories in the database (a special training set or data in which we want to detect outliers) after the monitored area is partitioned. As a result, each cell is represented by such a vector.

Now, let A be a new trajectory that is to be tested. For each cell c of the grid the trajectory passes, the instant outlying score is computed. Assume that A has K directions in c. Then A can be represented by a direction vector $\mathbf{v}_c = (v_{a1}, v_{a2}, ..., v_{aK})$ where $v_{a1} \quad (k = 1, ...K)$ are the directions. The direction-based instant outlying score *OScoreDir* of *t* in *c* is computed as:

$$OScoreDir_{Ac} = 1 - \sum_{k=1}^{K} q_k \sum_{i=1}^{8} p_i * cos\angle(v_{Ak}, v_i), \qquad (2)$$

where $q_k = 1/K \quad (k = 1, ...K)$ are normalizing constants, $cos\angle(v_{tk}, v_i)$ is the cosine value of the angle between direction v_{Ak}, which is the k-th direction of A in c, and v_i, which is the center direction of the i-th direction sector of c as it is shown in Figure 5.

The outlying score can also be based on the density of trajectories. Assume that each cell c is represented by the number of trajectories from the database passing it d_c. Then the density-based outlying score of A in c is defined as:

$$OScoreDen_{Ac} = \begin{cases} s, & if\ d_c < \tau \\ 0 & otherwise, \end{cases} \qquad (3)$$

where a density score s is a penalty for a low density and τ is a density threshold.

The main idea of TOP-EYE is that abnormal behavior of a moving object is gradually reflected in its moving characteristics represented by its trajectory. Therefore, it is possible to detect the trajectory outliers in an evolving way combining the instant outlierness of the object with the influence of its prior movement. The authors introduce an exponential decay function $exp(-\lambda\Delta t)$ to control the influence. Here λ is a user-specified parameter that determines the decay rate and Δt is a time interval between the time for which the current instant outlying score is calculated and some time t in history of the object movement for which the instant outlying score was calculated before. Let t_0 be the initial time instant for which the instant outlying score obtained by Equation 2 is S_{t_0}, next time instant is t_1 and score S_{t_1} and so on. Then the evolving outlying score at time instant t_i is calculated as:

$$\begin{aligned} ESCore_{t_i} \quad = \quad & S_{t_i} + S_{t_{i-1}} * exp(-\lambda\Delta t_{i-1}) \\ & + S_{t_{i-2}} * exp(-\lambda\Delta t_{i-2}) + ... \\ & + S_{t_0} * exp(-\lambda\Delta t_0), \end{aligned} \qquad (4)$$

The authors show in [2] that the evolving outlying score as defined in Equation 4 both makes it possible to detect an trajectory outlier in its early stage and can be robust with respect to the accidental increase of the evolving score if the score threshold is properly set.

Because of promising properties of the TOP-EYE method we decided to prepare experiments on real-world data sets. We were mainly interested in its applicability for mining outlier trajectories in surveillance systems.

The algorithm and a user friendly application for testing was implemented in Java [11]. We carried out experiments with two data sets. The first one contained trajectories extracted from videos we use in our research and development of a multi-camera surveillance system SUNAR [1]. This dataset will be referred to as SUNAR data set. The second one was a Hurricane data set[1] which is one of datasets that were used in

[1]http://weather.unisys.com/hurricane/atlantic/

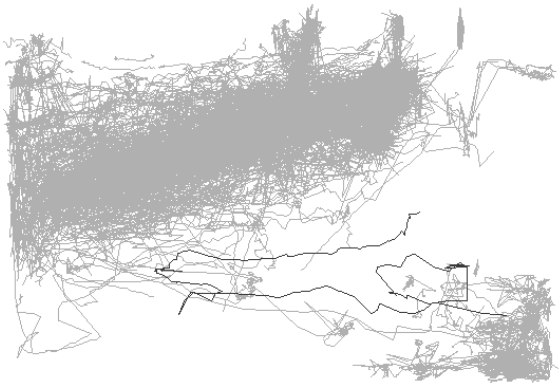


Fig. 6. Trajectory outlier detected in SUNAR data

Fig. 7. Hurricane direction-based trajectory outliers

Fig. 8. Hurricane density-based trajectory outliers

Fig. 9. Hurricane trajectory outliers detected by TRAOD. Adapted from [5]

experimental evaluation of the TRAOD algorithm mentioned in Section III-A. The comparison of results of TOP-EYE with results of another trajectory outlier detection algorithm was also one of our goals.

The SUNAR data set contained trajectories extracted from a set of videos from five cameras monitoring some space at the airport[2]. It emerged that the quality of extracted trajectories was in many cases low. The task of object detection and tracking in a monitored space where there are several people moving in various directions is difficult. When we compared the discovered trajectory outliers with the videos from which the trajectories were extracted, we have found that many of them are not real trajectories of one particular person. Instead, they were composed of segments of trajectories of several people. One of correct density-based trajectory outliers that represent the movement of only one person is shown in Figure 6. The parameters from Equations 3 and 4 were set to $\lambda = 0.4$, $s = 2$ and $\tau = 10$. The threshold of the evolving outlying score was set to 3.0 and the size of the grid was 50 units.

The Hurricane dataset that we used in experiments contained trajectories of Atlantic hurricanes from years 1950 to 2008. Each trajectory is described as a sequence of time, longitude and latitude values together with some meteorological data registered in six-hour intervals. The result of the direction-based trajectory outliers detection with $\lambda = 0.7$ is in Figure 7. The detected density-based trajectory outliers for parameters $\lambda = 0.7$, $s = 1$, $\tau = 10$ and the grid size of $5\,^\circ$ is shown in Figure 8. The thresholds for direction-based score and density-based score were set to 2.5 and 1,8 respectively.

Our Hurricane dataset that was very similar to the one used in [5] to evaluate the algorithm TRAOD. The result adapted from that paper is in Figure 9. It can be seen that TRAOD detects outlying partitions of the trajectories, not necessarily the whole trajectories.

We have also studied the influence of the algorithm parameter settings. The value of the decay rate λ makes it possible to control the influence of the historical trajectory outlierness on the current value of the evolving score. The number of discovered outliers decreases with increasing value of λ.

The density score s is a penalty for passing a grid cell with density below the density threshold τ. For a given threshold of the evolving score, the increasing value of s increases the number of detected outliers. The increase of τ also results in the increase of the number of detected outliers.

The influence of the grid cell size on the number of detected density-based trajectory outliers is illustrated in Figure 10. The bigger size is the less is the number of addends in Equation 4. This experiment was done with SUNAR data and values of

[2]http://scienceandresearch.homeoffice.gov.uk/hosdb/ cctv-imaging-technology/video-based-detection-systems/i-lids

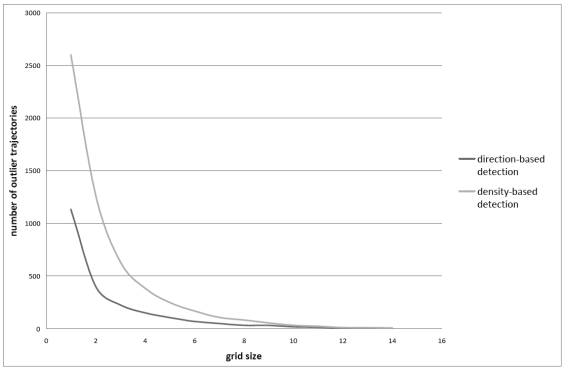


Fig. 10. The dependency of the number of detected outliers on the grid cell size

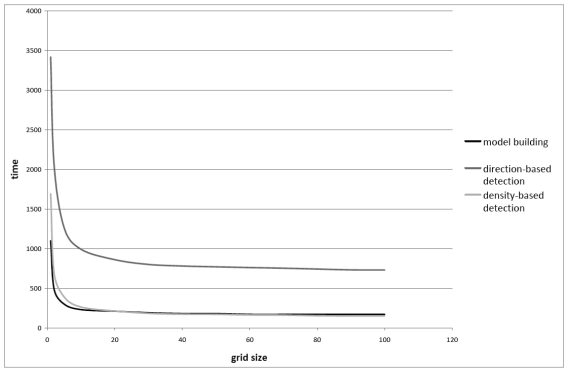


Fig. 11. The dependency of the processing time on the grid cell size

parameters $\lambda = 0.75$, $s = 1$ and $\tau = 5$. The evolving score threshold was set to 2.5.

Several other experiments were focused on time complexity, namely on the dependency of processing time on grid cell size, number of trajectories and total number of points of all trajectories . The values of parameters were set to $\lambda = 0.75$, $s = 1$ and $\tau = 5$ in all experiments. We measured both time of model building and time for both direction-based and density-based outlier detection. Figure 11 illustrates the dependency on the grid cell size. The dependency of processing time on the number of trajectories is shown in Figure 12. The dependency on the total number of points was very similar. It is evident that there are no significant differences in time complexity between direction-based trajectory outlier detection and density-based one.

IV. Conclusion

Location-based services, which are recently popular, produce a lot of moving object trajectory data. Mining movement patterns of such objects and trajectory outlier detection are two important analysis tasks with real-world application potential. This paper introduces fundamental concepts related to them and presents several representative approaches and techniques to solve them. Moreover, our experimental results provided by the TOP-EYE trajectory outlier detection algorithm on real data are briefly discussed here.

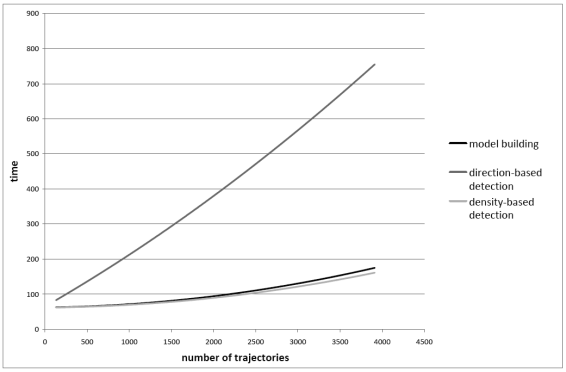


Fig. 12. The dependency of the number of detected outliers on the number of trajectories

Acknowledgment

The work was partially supported by the grant VG20102015006 of the Ministry of the Interior of the Czech Republic.

References

[1] P. Chmelar, A. Lanik and J. Mlich: SUNAR: Surveillance Network Augmented by Retrieval, In: ACIVS 2010, Sydney, AU, Springer, 2010, pp. 155-166.

[2] Y. Ge et al.: TOP-EYE: Top-k Evolving Trajectory Outlier Detection. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp. 1733 - 1736.

[3] J. Han, Z. Li, A.T. Tang: Mining Moving Object, Trajectory and Traffic Data. In: Kitagawa et al. (Eds.): DASFAA 2010, Part II, LNCS 5982, Springer-Verlag Brlin Heidelberg, 2010, pp. 485 - 486.

[4] J. Han, Z. Li, A.T. Tang: Mining Moving Object, Trajectory and Traffic Data. DASFAA 2010 Tutorial. Available on http://www.cs.uiuc.edu/homes/hanj/pdf/dasfaa10_han_tuto.pdf [September 2011].

[5] J.-G. Lee, J. Han, X. Li: Trajectory Outlier Detection: A Partition-and-Detect Framework. In: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, 2008, pp. 140 - 149.

[6] X. Li et al.: ROAM: Rule- and Motif-Based Anomaly Detection in Massive Moving Object Data Sets. In: Proceedings of the 7th SIAM International Conference on Data Mining, 2007, pp. 273 - 284.

[7] Z. Li el al.: MoveMine: Mining Moving Object Data for Discovery of Animal Movement Patterns. *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 4, July 2011, pp. 37:1 - 37:32.

[8] Z. Li et al.: Swarm: Mining Relaxed Temporal Moving Object Clusters. Proceedings of the VLDB Endownment, Vol. 3, No. 1, 2010, pp. 723 - 734.

[9] L. Liu et al.: Efficiently Mining Outliers From Trajectories of Unrestraint Movement. In: Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering, 2010, pp. V2-261 - V2-265.

[10] E.M. Knorr, R.T. Ng, V. Tucakov: Distance-based outliers: Algorithms and applications. *The VLDB Journal*, Vol. 8, No. 3 - 4, pp. 237-253, February 2000.

[11] M. Pesek: Knowledge discovery in spatio-temporal data, Master Thesis, Brno, FIT VUT Brno, 2011, p. 63.

Accelerator-Based Architectures for High-Performance Computing in Science and Engineering

Ivan Plander
A. Dubček University of Trenčín in Trenčín
Študentská 2, 91150 Trenčín
ivan.plander@tnuni.sk

Abstract - ***In the past years, a new class of high-performance computing (HPC) systems has emerged. These systems employ unconventional processor architecture – such as cell accelerators and graphics processing units (GPUs) - for heavy computations and use conventional central processing units (CPUs) mostly for non-compute-intensive tasks. General Purpose GPUs (GPGPUs) appear for scientific computing. A new concept is to use a GPGPU as a modified form of stream processor. The paper gives an overview of the state-of-the-art of the developments, applications and future trends in accelerator-based high-performance computing for all platforms: applications, hardware and software technologies, languages and development environments.***

Index Terms - *cell accelerator, graphics processing unit (GPU), general purpose GPU (GPGPU), high-performance computing (HPC), data parallelism, SIMD and SPMD models, CUDA programming model, GPU acceleration of matrix multiplication, green computer, flops per watt.*

I. INTRODUCTION

Today's communications technologies and scientific advances in computer hardware and software are forcing a dramatic change and acceleration in all areas of sciences and engineering. Many scientific applications require the acceleration of linear algebra operations for solution large problems, such as 100 000×100 000 matrix multiplication etc. In the past few years, a new class of high-performance computer (HPC) systems has emerged. These systems employ unconventional processor architectures – such as Cell accelerators [1] and graphics processing units (GPUs) [2] for heavy computations and use conventional central processing units (CPUs) mostly, for non-compute-intensive tasks, such as I/O and communication.

The aim of this paper is to give an overview of the state-of-the-art of the developments, applications and future trends in GPU-based high performance computing for all platforms: Applications, hardware and software technologies, languages and development environments.

Complexity of computing in science and engineering is rapidly increasing. In the past years complexity of simulations running on the HPC systems requires high processing speed, high level of parallelism (systems with thousands of processing elements), envisioned to reach millions of threads of parallelism and availability of parallelism in algorithms. Architecture problems to be solved for high-performance computing are:

(a) to state limits to manageable levels of parallelism and programming models allowing high performance and efficiency,

(b) to determine the number of cores that can be used to building a single computer and its heterogeneity (CPU/GPU),

(c) to specify the fundamental limits to increasing space dimensions of interconnect,

(d) design considerations for I/O and storage subsystems for huge amounts of data,

(e) to reduce as much as possible the power consumption but also enable the design of even faster computers

Architectures for extreme-scale computing use the following approaches:

(1) New key technologies: new near threshold voltage operations, non-silicon memories, photonics, 3D stacking, per-core efficient voltage and frequency management (key to energy and power efficiency),

(2) Fine-grained concurrency: efficient, scalable synchronization and communication primitives, hierarchical machine organization, processing-in-memory, resiliency (combined with failure tolerance), programming the machine with high-level data-parallel model, using intelligent compilers to map the code to the hardware,

(3) High-speed interconnect architecture for the next-generation supercomputers that operate beyond the petaflops limits. To improve high-performance computing systems by increasing the number of processor cores per node and nodes per system it must be used new architecture for interconnection networks employing mesh/torus topology of the 6D because of high scalability, fault-tolerant interconnection and low

cost/performance ratio.

Multi-core processors are no longer the future of computing they are only present day reality. A typical mass-produced CPU features multiple processor cores while a GPU (Graphics Processing unit) may have hundreds or even thousands of cores. GPUs in connection with CPUs create the heterogeneous architecture for high-performance computing. With the rise of multi-core architectures has come the need for advanced programming and for programming massively parallel processors.

II. GENERAL PURPOSE GRAPHICS PROCESSING UNIT

The Graphics Processing unit, or GPU, has been an integral part of most home computer systems and game consoles for several years. Efforts for ever more realistic games has driven its development from a simple 2D accelerator for graphics-based applications to an extremely powerful unit aimed at 3D games.

The raw computational power of the modern GPU has, in recent years, led to explosion of interest in its use for numerically intensive computing beyond the graphics domain. This interest is demonstrated by the release of dedicated General Purpose GPUs, or GPGPUs, by manufacturers such as NVIDIA) [2] and AMD) [3] The adoption of GPGPU computing by the HPC community is shown by the fact that two of the top four machines in the latest Top 500 list June 2011 employ GPGPUs.

A Graphics Processing Unit (GPU) is simply an accelerator, sometimes called a co-processor, designed to carry out specific graphics tasks faster than the main CPU in the system. It contains one are more microchips designed with a limited number of algorithms in mind. Graphics operations may be split into two types – vector- based operations and raster operations. Vector- based operations are the manipulation of the so called graphics primitives -- that is objects such as lines, circles and arcs. A raster, or bitmap, is a structure representing individual image pixels such as those displayed on scene. Raster operations are the manipulation of such bitmaps in various ways such as scrolling a background image between display frames or overlaying a moving sprite over a background.

GPUs were developed during the years 1970s and 1980s, however only in the mid 1980s the first mass-market personal computer appeared including a dedicated chipset capable of taking care of all the graphics functions. The chipset included the famous blitter chip named after the acronym for Block Image Transfer. The blitter was responsible for manipulating large amounts of data corresponding to bitmap images. As well as a personal computer being a popular machine for playing games the advanced graphics capabilities led to its use in video processing, production and scene rendering. However, only in the 1990s the development of GPUs began in earnest. It was in this period more advanced GPUs for IBM-compatible PCs started to be developed. The first of these were simple 2D accelerators, aimed at speeding up the performance of the user interface of the Windows operating system. At about the same time 3D computer games were starting to became popular, leading to the development of GPUs specifically aimed 3D graphics processing. These accelerators became available in games consoles and on the PC thanks to graphic cards. The race was on. Fuelled by the thirst for ever more realistic computer games, the development of 3D GPUs has continued apace ever since. Today market is largely dominated by two companies, AMD and NVIDA. Modern graphic cards are responsible for the many different operations involved in producing a graphic scene which are commonly referred to as rendering pipeline. The input to the pipeline is in the form of information about primitives, typically polygons or triangles. The rendering process transforms and shades the primitives and map them onto to screen for display. The typical pipeline steps are: Geometric vertex generation, Vertex processing, Primitive generation, Primitive processing, Fragment generation or rasterization, Fragment processing, Pixel operations or composition.

The rendering pipeline lends itself to a form of processing called stream processing. A stream of data is passed through a series computational stages. The operations within each stage are performed locally and independently on each element within the data stream. GPUs in such a way allow to exploit the inherent task-parallelism of streaming (different processor resources being devoted to different stages of the pipeline).

Several stages of the rendering pipeline also lend themselves naturally to another form of parallelism: data-parallelism. That means, each geometric vertex or image pixel can be processed independently of the others, but using the same algorithms, in the other words using the common Single Instruction Multiple Data (SIMD) approach. The data parallel approach has consequently evolved form SIMD to a more complicated Single Program Multiple Data (SPMD) model, as hardware capabilities have increased. In the SPMD model, different branches may be followed within a rendering stage for different sections of data.

Modern GPUs typically contain tens or even hundreds of processing units, each unit futher containing several Arithmetic Logic Units (ALUs) able to exploit the SIMD characteristic of much of the processing.

The fact that modern GPUs typically contain hundreds of ALUs leads to units of processing power over 1 Tflops, several times that of a typical CPU, for example NVIDIA GPU GeForce GTX580M [4]. This performance naturally led to an interest in using them for computationally-intensive problems outside the traditional graphics domain. Manufactures now

release products aimed specifically at the HPC market, in particular AMD FireStream [5] and the NVIDIA Tesla [2]. In the latest Top500 list of June 2011 [6] three of the fastest five machines employ GPGPUs. The top spot taken in this list is by the Japanese RIKEN supercomputer with a performance of 8,162 petaflops (1015) and second one the Chinese Tianhe-1A system with a performance 2,57 petaflops.

III. GENERAL PURPOSE GPU SCIENTIFIC COMPUTING

To make best use of the GPGPUs and achieve a performance which makes their use more worthwhile than a standard CPU in scientific computing, significant challenges must be addressed.

The primary issue is that of the application's scope for parallelism: it must demonstrate sufficient data-parallel characteristics such that it can be mapped on the GPU architecture and make full use of the many processing cores available.

The second challenge is making efficient usage of the GPU memory through the application's memory access patterns, where several problems may be encountered. The first problem to address is that of copying data between the main memory of the machine hosting GPU and device itself. This transfer is quite expensive therefore such transfers should be minimized whenever possible. Types of applications likely making the best use of GPGPUs: (a) Ensure that the application has substantial parallelism, (b) Ensure that it has high computational requirements, i.e. a high ratio of arithmetic operations to memory operations, (c) Prefer throughput over latency of memory access.

General purpose GPU computing is the use of a GPU to do general purpose scientific and engineering computing. The model for GPU computing is to use CPUs and GPUs together in a heterogeneous co-processing PU. The GPU has evolved over the years to have teraflops of floating point performance. NVIDIA revolutionized the GPGPU and accelerated computing world in 2006-2007 by introducing its new massively parallel architecture called CUDA. The CUDA architecture consists of hundreds of processor cores that operate together to crunch through the data set in the application.

Success of GPGPUs in the past few years has been easy of programming of the associated CUDA parallel programming model. In this model the application developers modify their application to take the compute-intensive kernels and map them to the GPU. The rest of application remains on the CPU. Mapping a function to the GPU involves rewriting the function to expose the parallelism in the function and adding C keywords to move data to and from the GPU. The developer is tasked with launching 10s of 1000s of threads simultaneously. The GPU hardware manages the threads and does threads scheduling.

The CUDA parallel hardware architecture accompanied by the CUDA parallel programming model provides a set of abstractions that enable expressing fine-grain and coarse-grain data and task parallelism. The programmer can choose to express parallelism in high-level languages, such as C, C++, Fortran or driver APIS, such as OpenCL [7] and Direct X-11 [8] (Fig. 1). The users should understand alike the

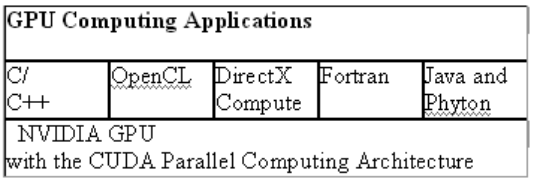


Fig. 1 Set of software development tools along with libraries and middleware

basic concepts of parallel programming and the GPU architecture. Case study demonstrates the development process, which begins with computational thinking and ends with efficient programs.

IV. GPU ACCELERATION OF LINEAR ALGEBRA OPERATIONS

Many scientific applications require the acceleration of linear algebra operations, which are quite well suited for GPU architectures. The CUDA software development toolkit includes an implementation of the basic linear algebra subprograms (BLAS) library ported to CUDA (Cublas). The most expensive operation is the matrix multiplication.

GPU acceleration of matrix multiplications can be realized using cleaving algorithm (9). Consider the matrix multiplication $\mathbf{C = A\,B}$, where $\mathbf{A}$ is an $(m \times k)$ matrix, $\mathbf{B}$ is a $(k \times n)$ matrix, and $\mathbf{C}$ is an $(m \times n)$ matrix. We can divide A into a column vector of $(r + 1)$ matrices

$$A = \begin{pmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{A}_r \end{pmatrix} \qquad (1)$$

where each entry Ai is a $(p_i \times k)$ matrix, and

$$\sum_{i}^{r} p_i = m. \qquad (2)$$

In practice, all the p_i will be the same.

In a similar manner, can be divided **B** into a row vector of $(s + 1)$ matrices $\mathbf{B} = (\mathbf{B}_0, \mathbf{B}_1, \ldots \mathbf{B}_s)$, where each $\mathbf{B}_j$ is a $(k \times q_j)$ matrix and

$$\sum_{j}^{s} q_j = n \qquad (3)$$

Again, all the q_j will be the same. Then the outer product of this two vectors can be formed:

C=

$$\begin{pmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{A}_r \end{pmatrix} \begin{pmatrix} \mathbf{B}_0 & \mathbf{B}_1 & \cdot & \cdot & \cdot & \mathbf{B}_s \end{pmatrix} = \qquad \mathbf{(4)}$$

$$\begin{pmatrix} \mathbf{A}_0\mathbf{B}_0 & \mathbf{A}_0\mathbf{B}_1 & \cdot & \cdot & \cdot & \mathbf{A}_0\mathbf{B}_s \\ \mathbf{A}_1\mathbf{B}_0 & \mathbf{A}_1\mathbf{B}_1 & \cdot & \cdot & \cdot & \mathbf{A}_1\mathbf{B}_s \\ \cdot & & \cdot & & & \cdot \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & & \cdot & \cdot \\ \mathbf{A}_r\mathbf{B}_0 & \mathbf{A}_r\mathbf{B}_1 & \cdot & \cdot & \cdot & \mathbf{A}_r\mathbf{B}_s \end{pmatrix}$$

Each individual element $\mathbf{C}_{ij} = \mathbf{A}_i \mathbf{B}_j$ is a $(p_i \times q_j)$ matrix, and can be computed *independently*. Generalizing this to a full implementation for the GPU we get a SIMD matrix multiplication. The p_i and q_j values can be chosen such that each submultiplication fits within the currently available GPU memory. Each multiplication can be organized through the GPU and assembled on the CPU.

Although the SIMD and SPMD models are supported by modern GPUs to make best use of hardware code-branching should still be minimized as much as possible. If *a* code branch occurs all threads must execute both branches, what is suboptimal for performance.

V. CONCLUSION

The main reason for using accelerators is because of the need to increase application performance to either decrease the compute time, increase the size of science problem that can be computed, or both.

However, reasons other than pure performance improvements are starting to influence the deployment of HPC resources. As the size of conventional HPC systems increases, their space and power requirements and operational cost quickly outgrow the available resources and budgets. Thus, metrics such as flops per machine, flops per watt of power, or flops per dollar spent on the hardware and its operation are becoming increasingly important. Accelerator-based HPC system look relatively attractive considering this metrics. The June 2011 Green 500 List of the world´s most energy efficient supercomputers shows the ranking in [10].

As a multi-billion dollar industry, the commodity games market will continue to be main driver of GPU development. Despite hardware systems (GPU-based high-performance computers) availability, however, the computational science community is currently split between early adopters of accelerators and skeptics. The early adopters´ main concern is that new computing technologies are introduced frequently, and users simply do not have time to chase after developments that might fade away quickly. With introducing application accelerators, new languages and programming models are emerging that eliminate the option to port code between „standard“ and „non-standard“ architectures. The community fears that these new architectures will result in the creation of many code branches that are not compatible or portable.

On the other side GPUs have evolved to the point where many real-world applications are easily implemented on them and run significantly faster than on the multi-core systems. Future computing architecture will be hybrid systems with parallel-core GPUs working in tandem with multi-core CPUs.

REFERENCES

[1] Kahle, J.A. et al.: Introduction to the Cell Multiprocessor. IBM J. Research and Development , vol.49, Nos.4-5, pp. 589 – 604 (2005)

[2] www.nvidia/object/tesla_computing_ solutions.html

[3] www.amd.com

[4] www.GeForceGTX570M.com

[5] www.amd.com/us/products/workstation/ firestream.aspx

[6] www.top500 June 2011.org

[7] www.OpenCL.apple.com

[8] www.DirectX11.microsoft.com

[9] Watson, M.A.: Accelerating Correlated Quantum Chemistry Calculation Using Graphics Processing Units. Computing in Science and Engineering, July/August, pp.40-50 (2010)

[10] www.The Green500 List - June 2011.org

Distribution of Manhattan Distances Between Processors within Two-Dimensional Grids

Zbigniew Domański
Institute of Mathematics
Czestochowa University of Technology
PL-42201 Czestochowa, Poland
zbigniew.domanski@im.pcz.pl

Abstract—**Grids of processors, radar arrays as well as wireless sensor networks are examples of two-dimensional network. The sum of the communication hops between the processors allocated to a given task influences the running time required to complete the task and thus, this quantity should be optimal. We study the statistics of pair-wise distances between processors, i.e. the number of communication hops, within certain two-dimensional tessellations.**

Keywords-component; distance distribution; two-dimensional grids

I. INTRODUCTION

Many questions lead to a problem of analysis of properties of distances distributions on regular networks [1,2]. Examples include, but are not limited to, material science or biology. For instance, in the field of computer science an important problem concerns the allocation of processors to parallel tasks in a grid of a large number of processors. This problem relays on the nontrivial correlation between the sum of the pair-wise distances between the processors allocated to a given task and the time required to complete the task [3].

The common question of the above mentioned problems is how many pairs of points separated by a given number q of steps can be found in a bounded region of a two-dimensional lattice. Such number q is referred to as the so-called Manhattan distance. More specifically, because the distance should be measured in terms of process and its activities, therefore functional distance should take into account the symmetry of the underlying lattice. A distance measure that accounts for this symmetry can be constructed around the $p-$ norm

$$\|\mathbf{x}\|_p = \left(\sum_{i=1,..,n} |x_i|^p\right)^{1/p}. \quad (1)$$

For $p=2$ we have the familiar Euclidean norm and for $p=1$ we get the Manhattan norm also called the taxicab norm. Thus, for a square lattice, the Manhattan distance is defined as the sum of the horizontal and the vertical distances. Similarly, for a given lattice, we can define the Manhattan distance as the sum of the distances along directions parallel to the edges of the lattice. This definition is equivalent to the definition of the distance between nodes in the graph that represents the lattice, i.e. the distance between two nodes u and v in a graph is the length of the shortest path from u to v.

From the mathematical point of view a grid of processors can be represented by the nodes of an appropriate flat lattice. In this paper we analyze the lattices which have edges and vertices formed by a regular tiling of a plane [4], so that all corners are equivalent, see Fig (1).

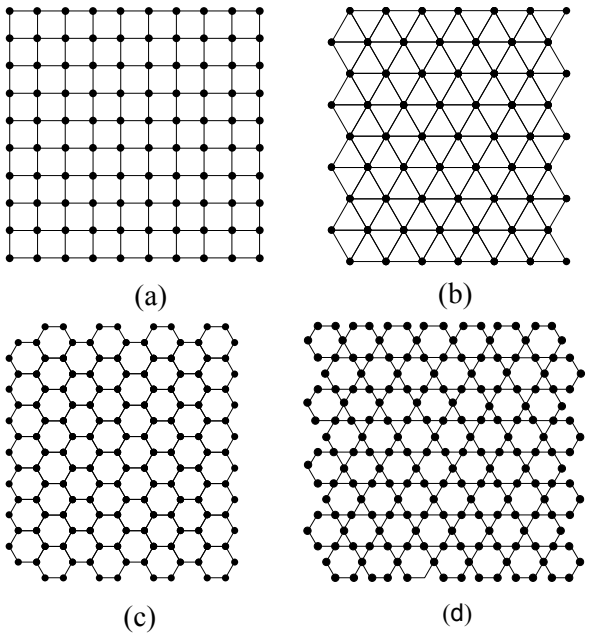


Fig. 1. Two dimensional lattices used in this work. They are represented by tessellations with: (a) square, (b) triangular, (c) hexagonal, and (d) Kagomé symmetries. For all tessellations processors are identified by the nodes. The edges form the shortest allowed paths between the pairs of processors.

II. GRAPH THEORY APPROACH

Graphs are useful for representing networks. In this section, we briefly present some definitions and background on graph theory and method that we use to count the pairs of grid's

nodes separated by a given value of the Manhattan distance. This question belongs to an ample class of the combinatorial properties of lattices. It is efficient to study such properties by using tools provided by the graph theory. To do this we map a grid onto finite, connected graph $G(V,E)$ whose vertices (nodes) $v_{i=1,\ldots,n} \in V$ represent processors, n is the number of vertices in G. Two vertices are adjacent if they are connected by an edge in E. A walk is a sequence of vertices each of which is adjacent to the previous one. If all vertices are distinct the walk is called a path. The length of a path is the sum of the lengths of all component edges. Since our graph represents the regular lattice then all its edges have the same length and, consequently, the path length is given by the number of visited edges.

An useful concept in the graph theory is the correspondence between graphs and so-called adjacency matrices, sometimes called connectivity matrices. An adjacency matrix $\mathbf{A}_G$ of G is the $n \times n$ matrix whose entries $(\mathbf{A}_G)_{ij} = 1$ if v_i and v_j are adjacent and zero otherwise. Adjacency matrix is very convenient to work with. For instance, let $\mathbf{A}_G^k = \mathbf{A}_G \cdot \ldots \cdot \mathbf{A}_G$ be the k-times matrix product of $\mathbf{A}_G$, then $\left(\mathbf{A}_G^k\right)_{ij}$ is the number of walks of the length k from v_i to v_j in G.

Our approach consists of two steps: (i) with the help of the family of matrices $\mathbf{A}_G^k$, each pair of nodes is assigned the smallest value of k so that the corresponding entry of $\mathbf{A}_G^k$ is nonzero. (ii) for each value of k we count the number of pair of nodes related to this value. Since the graph is finite, this approach yields a partition of Manhattan distances.

III. Results

We present the detailed calculations of distance distributions for four tessellations:

A. Square lattice

First we analyze the square lattice of processors with the lattice constant $a \equiv 1$. Without loss of generality, let us assume that the grid has the shape of a square whose side contains L nodes. Thus, the maximum value of the processor-to-processor distance $q_{\max} = 2(L-1)$ corresponds to two pairs of nodes located in the opposite corners of the grid. On the other hand $q_{\min} = 1$ is related to the number of pairs of nodes connected by edges of the grid. Each of the L rows and columns contain $L-1$ edges and this means that there are $2L(L-1)$ such distances. Following the approach described in section II we obtain an expression that describes the number $N(L,q)$ of distances q within the square-shaped grid

$$N(L,q) = \begin{cases} 2Lq(L-q) + \frac{1}{3}(q^2-1)q, & q \le L, \\ \frac{1}{3}\left[(2L-q)^2 - 1\right](2L-q), & L < q \le 2(L-1). \end{cases} \quad (2)$$

In Fig. (2) we show $N(L,q)$ for different values of L.

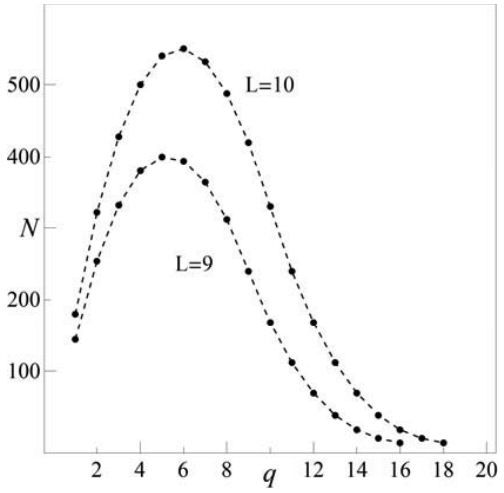


Fig. 2. Distribution (2) of the processor-processor distance for the square lattice related to two values of grid's length L. The line are drawn using (2) and they are only visual guides.

Equation (2) can be written in the form of the probability distribution function of distance with the help of the normalization condition, namely

$$P(L,q) = \frac{2}{L^2\left(L^2-1\right)} N(L,q). \quad (3)$$

B. Triangular lattice

Here and in the following subsections we analyze the triangle-shaped networks: the triangular, the hexagonal and the Kagomé lattices. They are built around the same set of nodes, see Fig. 3, and this will enable us to directly compare the results. Our approach applied to the graph represented in Fig. 3(a) yields the distance distribution in the form

$$N(L,q) = \frac{3}{2} q(L-q)(L-q+1), \quad q = 1,2\ldots,L-1. \quad (4)$$

Thus, the corresponding probability distribution of distances $P(L,q)$ is as follows

$$P(L,q) = \frac{12}{\left(L^2-1\right)L(L+2)} N(L,q), \quad (5)$$

The lattice size $L \geq 2$ is shown in Fig. 3(c).

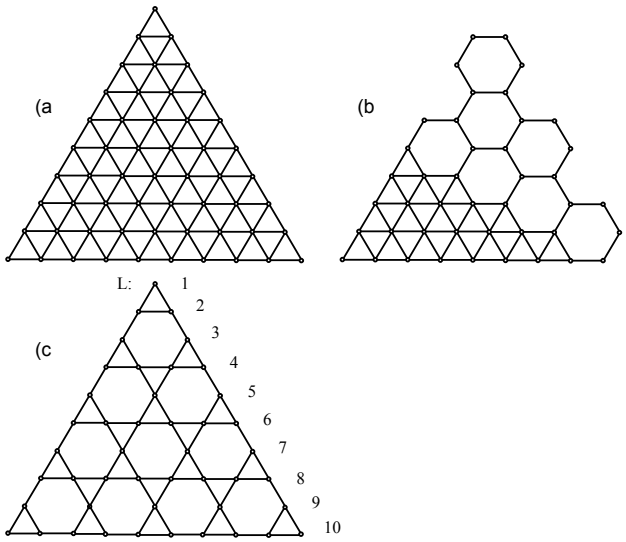


Fig. 3. Finite triangular lattice (a) viewed as an undirected graph. Sub-graphs of this graph represent the hexagonal lattice (b) and the Kagomé lattice (c). In part (b) it is seen how the hexagonal lattice emerges from the triangular one. The lattice size L is shown in (c).

C. Hexagonal lattice

The hexagonal lattice, see Fig. 3(b), viewed as a graph, possess fewer nodes and edges than the graph of the underlying triangular lattice presented in Fig. 3(a). Thus, within the same support, all functions representing the hexagonal symmetry take smaller values than these related to the triangular symmetry. The functions appropriate to the hexagonal symmetry read

$$P(L,q) = \frac{32}{(L-2)(L+6)\left(L^2+4L-8\right)} N(L,q), \qquad (6)$$

where

$$N(L,q) = \frac{3}{8} Lq(L-2q+4) - 3 + \frac{3}{8} q \left[q + \frac{1-(-1)^q}{2} \right] (q-4), \quad q = 1,\ldots,L-1. \qquad (7)$$

The lattice size $L \geq 4$ is shown in Fig. 3(c).

D. Kagomé lattice

Arguments, similar to these stated in the case of the hexagonal symmetry, make the relevant functions defined as follows

$$N(L,q) = \begin{cases} \frac{3}{4}L^2, & q = 1, \\ \frac{3}{16}(5q-2)(L-q)(L-q+2), & q = 2,\ldots,L-2, \\ \frac{3}{8}(L+1-q)\left[(2q+1)L+3-q(2q-1)\right], & q = 3,\ldots,L-1. \end{cases} \qquad (8)$$

and

$$P(L,q) = \frac{128}{3L(L+2)\left(3L^2+6L-8\right)} N(L,q). \qquad (9)$$

The lattice size $L \geq 2$ is shown in Fig. 3(c).

IV. Summary

In this paper we have derived the distributions of distances for the following grid's symmetries: square, triangular, hexagonal and Kagomé. These functions are polynomials of at most the third degree in the grid-node-concentrations and they yield the probability weight of class q containing pairs of nodes with given distance q. Our results could serve as an initial class of distribution functions of the node-to-node distance. In different context, some of the above mentioned results, concerning the square and the triangular symmetries, have been obtained elsewhere [5].

This paper focuses on geometry but the knowledge of the Manhattan distances in a particular lattice can be useful for studying many quantities of technological importance.

References

[1] E. J. Janse van Rensburg, "Statistical mechanics of directed models of polymer in the square lattice", Journal of Phys. A: Math. Gen., 2003, vol. 36(15), pp. R11-R61.

[2] C. M. Bender, M. A. Bender, E. D. Demaine, and S. P. Fekete, „What is the optimal shape of a city?", Journal of Phys. A: Math. Gen., 2004, vol. 37(1), pp. 147-159.

[3] V. J. Leung, et. all, "Processor allocation on Cplant : Achieving general processor lacality using one-dimensional allocation strategies", Proceedings of the 4th IEEE Int. Conference on Cluster Computing, Willey-Computer Society Press, Chicago, 2002, pp. 296-304.

[4] B. Grünbaum, G. Shepard, "Tilings and Patterns", New York, W. H. Freeman, 1986.

[5] Z. Domański, "Geometry-Induced Transport Properties of Two Dimensional Networks", in: "Advances in Computer Science and Engineering", M. Schmidt (ed.), 2011, pp. 337-352, InTech, Rijeka. Available: www.intechweb.org/books.

Embarrassingly parallel problem processed on accelerated multi-level parallel architecture

Miloš Očkay
Department of Informatics
Armed Forces Academy of gen. M. R. Štefánik
Liptovský Mikuláš, Slovakia
milos.ockay@aos.sk

Martin Droppa
ICT Department
Armed Forces Academy of gen. M. R. Štefánik
Liptovský Mikuláš, Slovakia
martin.droppa@aos.sk

Abstract— **Embarrassingly parallel problems require little or no communication among the processed tasks. They are easily adaptable on parallel architectures. In this paper we present issues related to embarrassingly parallel image processing on accelerated multi-level parallel architecture. As a part of this paper we created program for composition of stereoscopic sequence of images using CPU and GPU platform. Both solutions have been tested on the large datasets containing thousands of high definition images. Obtained results clearly distinguish pros and cons of introduced solutions.**

Keywords: CPU, CUDA, GPGPU, GPU, Embarrassingly parallel, Parallel, Cluster, MPI, Stereoscopy, Anaglyph, General-purpose computing, Compute unified device architecture.

I. INTRODUCTION

Amdahl's law offers the possibility to predict the speed-up of the problem processed on the parallel architecture. Problems comprised of parallel and sequential partitions have to deal with limited speed-up possibilities. Sequential partitions of processed problem usually limit the parallel processing and allow us only use limited number of parallel computational resources. On the other hand, embarrassingly parallel problem (epp) has no or very small sequential partition. This sort of problems can be very well parallelized and distributed to the parallel architecture for processing. It is possible to achieve significant speed-up and scalability of the problem. However, it is not directly implied that embarrassingly parallel problem will exceed speed of the sequential solution. There are a few important facts which have to be pointed up.

II. EMBARASSINGLY PARALLEL PROBLEM

Embarrassingly parallel problem can be defined as follows. There is no sequential partition included in the problem (if there is the one, parallel partition is not dependant of its output). Pure function *f* takes the input *x* and produces output *y*. The term *pure* describes the fact that function only modifies local state. There is no communication or negotiation of variables with the outside world. Data dependencies and the bondings can still exist, but they are defined on the same stage and, they keep the processing local. Data dependency creates data partitions (Fig. 1). Data in the partition have to use the same computational resource to keep processing local [1]. Actual data element in the partition usually uses neighborhood elements to compute current value. The proportion of the partition is defined by the dependency, and its locality is usually limited by the local memory size.

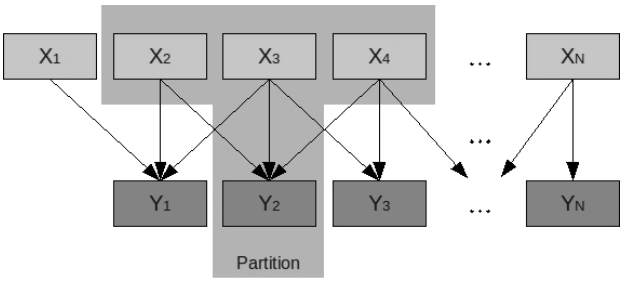


Figure 1. Data partitions in epp

III. MULTILEVEL PARALLEL ARCHITECTURE

Computer cluster is a well known architecture used for the parallel processing. Scalability of this architecture is represented by the ability to add or remove compute nodes [2]. Bottleneck of this architecture is interconnection among the nodes and it usually requires expensive hardware to achieve high speed, throughput and low latency parameters.

There is also a different approach how to scale this architecture up. Each cluster node can be equipped with the graphics accelerator. Graphics accelerator includes its own massively parallel processor which is called Graphics processing unit (GPU). Nowadays, this processor can be used for general purpose computing [3]. With this approach, computer cluster can include thousands of parallel processors within the small range of compute nodes. The cluster architecture which includes graphics accelerators can be called an accelerated computer cluster. An accelerated computer cluster is a multistage (multilevel) parallel architecture. Multiple parallel stages can be distinguished and they offer greater possibilities in the process of decomposition. The basic cluster stage includes the group of compute nodes with one or multiple CPUs. The node stage includes specific compute node (host system) which can be expanded by the accelerator stage (Fig. 2). Communication between host system and accelerator is based on PCI Express bus. For more detailed description of

accelerated host system, we recommend the further study of the graphics accelerators manufacturers' specification (NVIDIA, AMD).

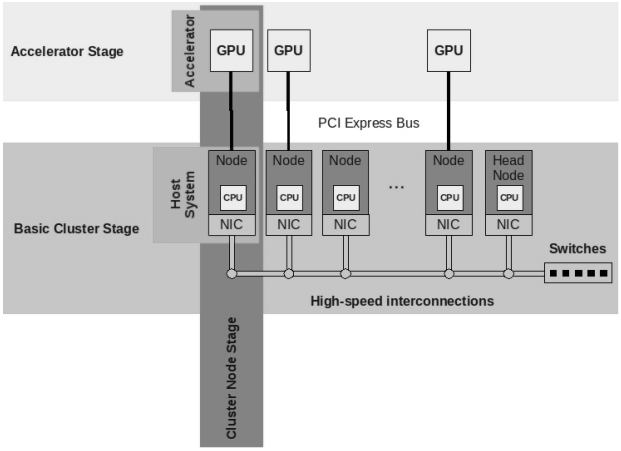


Figure 2. Acclelerated cluster architecture

IV. STEREOSCOPIC SEQUENCE OF IMAGES

Stereoscopic video sequence generation is an excellent example of embarrassingly parallel problem. The video is created as the sequence of frames. The video sequence contains thousands of frames. Two video sequences are generated for stereoscopic video (anaglyph). Sequences for left and right eye use the different view angle of the scene. Because the display devices use only one display, right eye and left eye sequence separation is done as follows. Each frame in both sequences is filtered with chromatic filters. Filters for right eye and left eye frames are different. Filtered frames are then combined to the final stereoscopic frame. Stereoscopic frames are combined to the stereoscopic video sequence. Stereoscopic sequence can be viewed with stereoscopic anaglyph glasses, thus the viewer can experience the feeling of depth in image (Fig. 3).

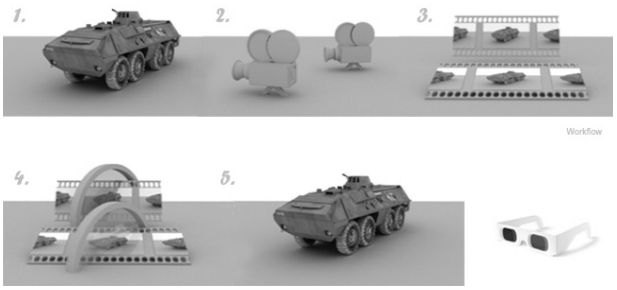


Figure 3. Stereoscopic video sequence generation

It is possible to divide this problem into the multiple computationally intensive parts. Firstly, the actual frames for the left and right eye are generated in the process of rendering the scene by the 3d application. The result of this process is the sequence of frames for the left and the sequence of frames for the right eye. Each frame consists of pixels. The number of pixels in frame defines the quality of the output image. We used 720p and 1080p resolution for the problem testing. RGBA model with 8 bits included in each channel was used for the pixel color definition. The frame rate was set to 25 frames per second for each eye sequence. According to these parameters it is possible to determine the size of processed dataset. The following example shows the dataset size calculation for stereoscopic video based on 1080p resolution and frame rate set to 25 frames per seconds. The length of video is 30 seconds (1).

$$D = t * f * \{2(res * RGBA / 1024^2)\} \quad (1)$$

$$D = 30*25*\{2(1920*1080*32/8)/1024^2\} = 10\,500 \text{ MB}$$

D-dataset size [MBytes], t-length of video [seconds], f-frame rate [prames per second], res-resolution [pixels], RGBA-number of bits per pixel.

A closer look at the dataset shows that it is a trivial form of embarrassingly parallel problem. Sequences for the left and right eye are comprised of independent frames. It is not necessary to process these sequences in sequential fashion. It is possible to split the sequence to multiple smaller sequences and process them concurrently. This fact is very useful in the process of rendering frames. Partial sequences can be separated to standalone frames. Because one frame is not dependant on another one, it is possible to process them in parallel. The order of frames is important and can be disorganized by concurrent processing. The order can be preserved by the clever naming convention of the frames. Pixels in frame are also independent and can be processed in parallel. There is only one restriction which has to be deliberated. It is a data locality for pertaining left and right eye frames. These frames have to be collocated in the same memory space and processed with the same computational resource [4].

V. DECOMPOSITION

This problem was processed on the multilevel accelerated architecture described in the section III. In the process of decomposition, dataset was divided accordingly to make a valid use of all accelerated cluster stages. On the basic cluster stage, the decomposition is represented by the horizontal splitting of video sequence to the smaller sequences. Video sequence cannot be split vertically because of data locality of pertaining frames. Cluster rendering was used for producing frames of partial sequences. 3D application only distributes parameters of scene across the cluster nodes [5]. The rendering is done locally on each concerned node. The fact that only parameters of scene are distributed saves a lot of transfer time. In the case of local rendering, partial sequences have to be distributed to the nodes and according to massive size of the input it would be a wasteful approach. The rendering is done on the basic stage of the cluster and accelerator is not involved in the computation.

The next step is processed on the cluster node stage (Fig. 4). Each node processes partial sequences. This process

includes filtering of the left and right eye frames and the final composition of these frames. Accelerator (accelerator stage) was used for the filtering and final composition. CPU also was used for a comparative purpose (Fig. 5). CPU processing was sequential and results for the GPU and CPU version are compared in the next section of this paper.

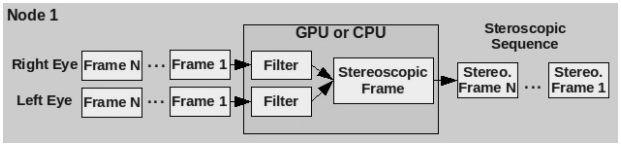


Figure 4. Frame processing

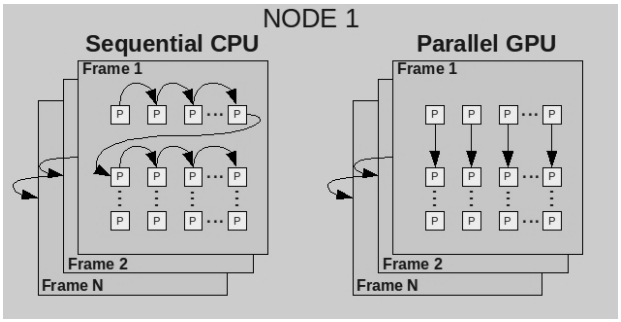


Figure 5. Sequential CPU vs. parallel GPU

VI. RESULTS

We created Linux CUDA based program to illustrate accelerated processing of embarrassingly parallel problem. Please follow the URL in references section [6] to download the program for free. The program uses GPU and CPU and applies the left and right eye filters and final composition of stereoscopic frames. The program follows the computational structure described in section V. Multiple stereoscopic anaglyph schemes are supported. The program can create the output for various stereoscopic glasses. Two massive datasets were rendered for the testing purpose. The first dataset contained 10 000 frames with 720p resolution for left and right eye. The second dataset differs in resolution which was 1080p. The size of the larger dataset was around 500 GB. Results were achieved with Red-Cyan anaglyph scheme. It is the most common scheme in stereoscopic video processing. We only present results for 1080p because of paper size limitations. However, results for 720p dataset share the same conclusions. Results show not only independent GPU processing times but also times which include PCI Express transfers. CPU times are shown for comparative purpose.

Speed-up of GPU solution is not obvious for small number of frames (Fig. 6). Powerful CPU can still process few frames with the time value comparable to the parallel GPU. Results are changing rapidly with rising number of frames. Speed-up of GPU is represented by the division of CPU and GPU time values. As Fig. 6 shows, speed-up value for GPU has the increasing trend. Steepness of the curve represents scalability of solutions. GPU solution is scaling very well. On the other hand, scalability of CPU solution is poor. The shape of the curve plays an important part as well. Achieved results show that shape of curve is almost a linear line. It is possible to add another accelerated node and process twice as many frames as with one node. Processing time stays the same. This is also true for any number of additional nodes. These facts validate the possibility of embarrassingly parallel problem to achieve massive speed-up and good characteristic of scalability.

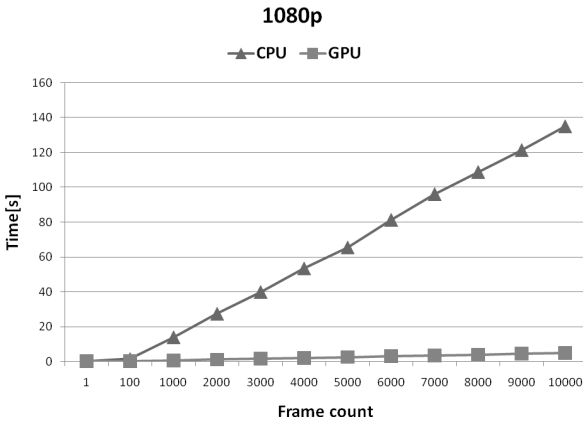


Figure 6. CPU and GPU without transfers

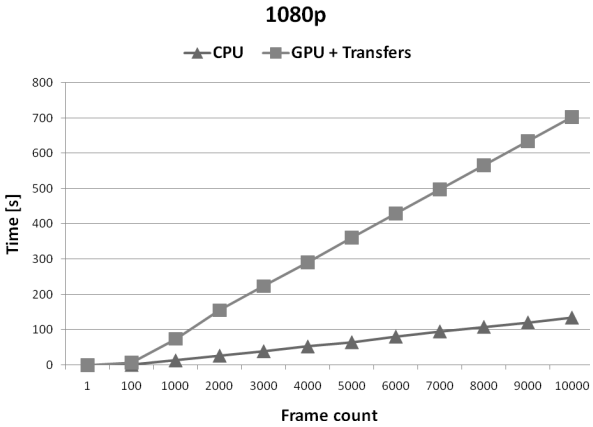


Figure 7. CPU and GPU + transfers

Important changes take place if PCI Express memory transfers are involved. These transfers allow the transport of data between a host system and an accelerator. Fig. 7 shows that transfers degrade parallel computation. In this case parallel computation does not even remotely reach the quality of the CPU computation. Each frame requires transfer of many megabytes over the PCI Express bus. In this case it is not possible to combine data to use the advantage of the optimization mechanisms like shared memory. It is also not possible to hide transfer latency with the computation because

the ratio between transfers and computation is too high. Transport has to be performed after every frame and it has to transfer always the same amount of data. The problem we presented is embarrassingly parallel, but memory transfers spoil the final results and the parallel computation is not useful in this case.

VII. HIDING TRANSFERS WITH COMPLEXITY

Important dependency facts can be concluded according to the results of previous tests. There is a connection between the size of transported data and complexity (*K*) of the computation. Complexity can be defined as the division of sequential (t_S) and parallel time (t_P) of computation.

$$K = t_S / t_P \quad (2)$$

The bigger K value, the bigger is the advantage of the parallel solution over sequential one. Values from interval [0,1] show that it is not reasonable to use accelerator for the computation. K value is decisive only if its value falls outside the interval [0,1]. In this case, PCI Express transfer time (t_T) has to be taken to consideration. t_T includes transfers of input data and results over the PCI Express bus.

$$t_S > t_P + t_T \quad (3)$$

Parallel solution advantage can be improved by the following ways. t_T has to be as small as possible. In presented case it can be done by the reduction in resolution of the frames or by the reduction of the overall frame rate. Concerning embarrassingly parallel problem t_P value is usually much lower than t_S value. Parallel solution is successful only if (3) is true (Fig. 9). It is obvious how to decrease t_T value. Low value of t_P and high value of t_S implies high value of K. If there is a high value of K and the low value of t_T it is possible to hide transfers with computation. The question is what affects parameter K and what improves this parameter? Values t_S and t_P rise with number of performed mathematical operations. It is important that the values t_S and t_P increase in a different rate. We should have a closer look at the previous problem focusing on number of operations performed on pixels. t_S and t_P values can be evaluated as follows:

t_s = Number of operations * Number of pixels * Time per pixel operation

t_P = Number of operations . Time per pixel operation

The parameter "Number of pixel" is omitted in parallel computation. Many processors are involved in computation and pixels are processed concurrently. The number of concurrent tasks depends on number of processors, number of pixels and configuration of kernel [7]. Addition of one operation per pixel will only prolong t_S a bit and a sequential solution will be affected heavily (Fig 8).

The number of operations performed on single item of embarrassingly parallel problem along with PCI Express transfers are important factors which determine the success or failure of accelerated solution.

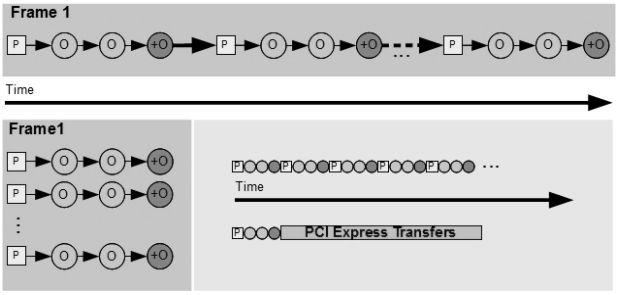


Figure 8. Komplexity and time dependecies P-pixel, O-operation

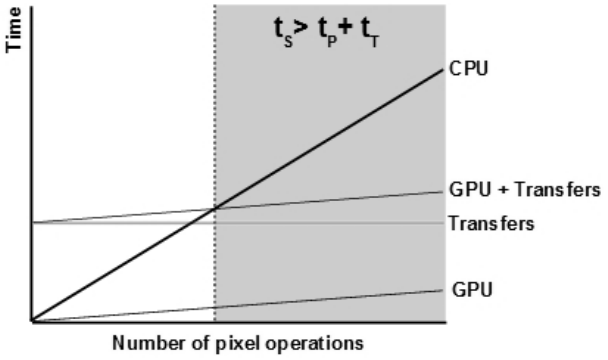


Figure 9. Hiding transfers

VIII. CONCLUSIONS

Accelerated computer cluster is a powerful computational tool for solving parallel problems. However, it is linked with the problems which have specific dataset characteristic. Embarrassingly parallel problem can use this architecture efficiently, but there has to be set the balance between the computation and the transfers. New accelerated processing unit (APU) is a promising technology which avoids transfers between host system and graphics accelerator by placing both CPU and GPU on a chip.

REFERENCES

[1] J. Dongarra, I. Foster, G. Fox, W. Gropp, K. Kennedy, L. Torczon, A. White, "Sourcebook of parallel computing," Elsevier. San Francisco, 2003.

[2] A. Vrenios, "Linux Cluster Architecture," SAMS. Indianapolis, 2002.

[3] N. Ganesan, M. Taufer, "Reformulating Algorithms for the GPU," GPU Technology Conference, 2010.

[4] D. B. Kirk, Wen-mei W. Hwu, "Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series)," Morgan Kaufmann. Boston, February 2010.

[5] H. Ayala, "GPUs at the Computer Animation Studio," GPU Technology Conference, 2010.

[6] M. Ockay, "Anaglyph image composition for high definition video," Harvard University - CS264, spring term 2011. www: http://cs264.mnemonix3d.com

[7] J. Sanders, E. Kandrot, "CUDA by exampele: An introduction to General-Purpose GPU programming," NVIDIA. Boston, July 2010.

Measurement of Network Traffic Time Parameters

Juraj Giertl*, Ľuboš Husivarga†, Martin Révés‡
Adrián Pekár§, Peter Feciľak¶
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice, Slovak Republic
Email: *juraj.giertl@tuke.sk, †lubos.husivarga@gmail.com, ‡martin.reves@tuke.sk
§adrian.pekar@tuke.sk, ¶peter.fecilak@tuke.sk

***Abstract*—The paper deals with the measurement of time parameters of network traffic using the BasicMeter tool developed in conformity with IPFIX standard. Method for one-way delay measurement is proposed, including time synchronization of metering points, compensation of clocks' skew, and transfer of timestamps for the observed packets. Experimental verification of the method is presented as well.**

I. Introduction

Delay is one of the major parameters of network traffic characteristics. Compared to other delays, one-way delay (OWD) provides more information than the round trip time (RTT). On the other side, measurement of one-way delay is much more difficult [1].

One-way delay expresses time difference between sending a packet from one endpoint and receiving it in another endpoint. This delay can be divided into several parts:

- propagation delay - the time needed for transfer the packet through physical medium
- processing delay - the time needed for packet processing (i.e. router, terminal equipment)
- queuing delay - time that a packet spends waiting in queue (i.e. before dispatch, before processing)

Propagation delay should be constant for a particular route, it depends on the line capacity and its length. However, processing and queuing delays are variable. They depend on current utilization of active network components by different types of traffic [2]. Measurement of each parts of delay is not impossible, but it would be very expensive. Therefore, we consider one-way delay as a whole. We can do some derivations from measured statistical sample of one-way delay. The minimum value would be approximately propagation delay, variation would be processing and queuing delay.

The meaning of direct measurement of one-way delay is important for several reasons. If we want to ensure the quality of services, it is important for us to know the characteristics of individual sections of our network. According to lines utilization, we can adjust and adapt network devices. For example, if we have a point in the network, which is congested and increases the delay, in such a place we can increase the capacity of lines [3], increase capacity of queues [4] or re-route part of the network traffic outside this area. This way we can ensure a better quality of service.

Not only the network devices, but as well as user applications can adapt to the load of computer networks. For example, an application transferring video or audio via network can adjust the quality of transmitted media by increasing or decreasing, depending on the current line utilization. They can thereby avoid unpleasant tearing of video or audio.

A. One-Way Delay Measurement

The principle of measurement is simple. Quite a serious problem is to achieve the appropriate circumstances. Let's have two endpoints of a network. One-way delay between them is measured, that one end point sends towards another a message, which will contain actual timestamp of message departure. In the second endpoint message is received and record the timestamp of its arrival. The difference between timestamps represents a one-way delay, but only if the clock at both end points are synchronized. This circumstance is not easy to achieve.

II. Scheme of One-Way Delay Measurement in BasicMeter Tool

The problem is described method for one-way delay measurement and the impossibility of its implementation in BasicMeter architecture. If we measure one-way delay between two points on the network, we are sending a message from one metering point to another. This is inadmissible in our architecture. Sending messages from the metering point would be violation of principles of passive measurement. Therefore, it is necessary to propose a different method of measurement.

A. Metering Points Clock Synchronization

No current synchronization method satisfies our needs. We will try to propose our own one, the most accurate, easily applicable which also takes care of clock divergence and continuous synchronization in time.

Exporters are synchronized against the collector. Independently, we got the same way connecting the collector and exporter in case of sending timestamp and synchronization. This simplifies the deployment of our measurement platform BasicMeter. One connection from exporter to the collector is sufficient for everything we needed to measure one-way delay.

Before we get to the resulting algorithm, let's say something about two clocks divergence. Let's start with a theoretical description of the general clocks.

In this section we introduce the terminology we use to describe clock behavior. A clock is a piecewise continuous function that is twice differentiable except on a finite set of points:

$$C : R \longrightarrow R$$

where $C'(t) = dC(t)/dt$ and $C''(t) = d^2C(t)/dt^2$ exist everywhere except for $t \in P \subset R$ where $|P|$ is finite. [5]

A "true" clock reports "true" time at any moment, and runs at a constant rate. Let C_t denote the "true" clock; it is the identity function given below, [5]

$$C_t(t) = t \tag{1}$$

We use the following nomenclature from [6] and [7] to describe clock characteristics. Let C_a and C_b be two clocks: [5]

- offset: the difference between the time reported by a clock and the "true" time; the offset of C_a is $(C_a(t) - t)$. The offset of the clock C_a relative to C_b at time $t \geq 0$ is [5]

$$C_a(t) - C_b(t) \tag{2}$$

- frequency: the rate at which the clock progresses. The frequency at time t of C_a is [5]

$$C'_a(t) \tag{3}$$

- skew: the difference in the frequencies of a clock and the "true" clock. The skew of C_a relative to C_b at time t is [5]

$$(C'_a(t) - C'_b(t)) \tag{4}$$

Two clocks are said to be synchronized at a particular moment if both the relative offset and skew are zero. [5]

In our case we have two clocks. In addition, they are on the sender and receiver side, let's name it as follows:

- C_s - sender clocks
- C_r - receiver clocks

The behavior of clocks C_s and C_r at time t can be expressed as follows:

$$C_s(t) = k.t + q \tag{5}$$

$$C_r(t) = l.t + r \tag{6}$$

where k, l express clock skew against "true" clock and q, r express start value at time $t = 0$.

Next, we will express relative offset between our two clocks. Clocks offset C_s and C_r at time t by (2):

$$O(t) = C_s(t) - C_r(t) \tag{7}$$

After substituting (5) and (6) we get:

$$O(t) = (k.t + q) - (l.t + r) \tag{8}$$

after arrangement:

$$O(t) = (k\ -l).t + (q - r) \tag{9}$$

As we can see on (9), we have a linear function. From time scope it can be constant, increasing or decreasing, depending on whether the skew difference $(k - l)$ of clocks is equal or not.

If we want to synchronize two clocks, and therefore we want to compensate their asymmetry and whole offset, first we need to find out how they differ. We have performed a measurement. We were sending timestamp t_1 with actual time from metering point with clock C_r to metering point with clock C_s. We have made timestamp t_2 with actual time as arrival time for timestamp t_1. Then we made a subtraction $\Delta t = (t_2 - t_1)$. This means time offset between clocks C_s and C_r. We have noticed, that Δt values are constantly increasing, as we can see on Fig. 1.

Constantly increasing trend can be described with the linear function $y = a.x + b$. This linear function describes variable clock offset between C_s and C_r at time t. It is equal with function (9).

If we want to determine the parameters of the linear function, we must first obtain the changing value of the offset. The first way of measuring is not sufficient. The problem is that the time difference Δt is not the actual clock offset, but combined with the delay of direct connection of two network endpoints. We need to measure one-way delay between those endpoints and subtract it from Δt.

For this purpose, we have made new way of clock offset measurement. We will be sending timestamp t_0 with actual time from endpoint with C_s to endpoint with C_r. In second endpoint we will make arrival timestamp t_1 and send it back. When this message arrives to first endpoint, we will make another timestamp t_2. So we have three timestamps measured. One-way delay is $OWD = (t_2 - t_0)/2$. And real clock offset is $(t_2 - t_1) - OWD$.

After approximation of clock offset with linear function, we are getting two parameters a and b. Now we are able to calculate clock correction. We will calculate correction for clock C_s, so they will be synchronized against clock C_r. The correction is shown as follows:

$$C^s_s(t) = C_s(t) - (a.t + b) \tag{10}$$

Where C^s_s means synchronized clock on metering point side with clock C_s, a and b are parameters acquired via approximation.

B. The Proposal of Timestamp Transfer

We need to approach to method of measuring one-way delay. We will not send a message from the metering point. We can use existing network traffic instead. For direct measurement of one-way delay, we need two metering points. Therefore we assume that the network traffic is passing through those two points.

Assume IPv4 network traffic. Depending on the location of our two metering points, we have a chance of an IP packet passes two metering points. Then comes a situation like where one of the metering point has sent a message, and the second metering point has accepted it. Like, therefore, that the IP packet has not been generated by our metering points, but was part of the network traffic.

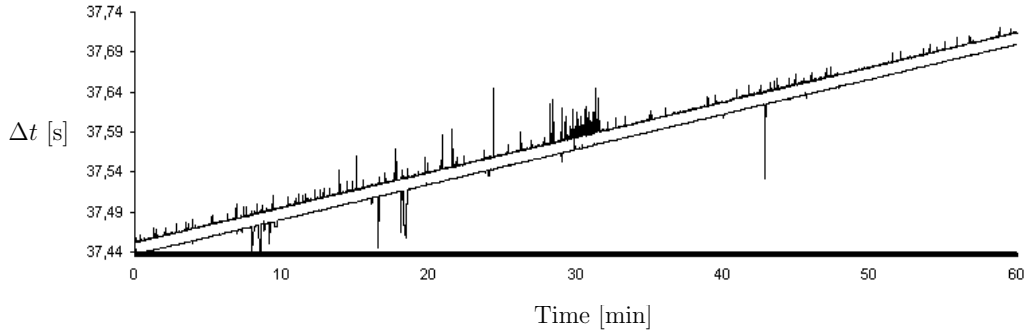


Fig. 1. Measured offset between two clocks containing one-way delay.

Since we cannot interfere with the measured network traffic, we must find another way to transfer timestamp. We are looking for sort of connection between two metering points, other than the direct connection between them. We had a look at BasicMeter architecture and noticed that each exporter (metering point) is linked to the collector. Possible path thus leads through the collector through which they can also exchange timestamps.

Exporter and collector are connected via IPFIX protocol [8]. Information links directs from the exporter to the collector. Opposite direction is not used. We have thus two routes from exporters towards the collector. To be able to say that we have a connection between the two metering points, both must be associated with one and the same collector.

Note that the connections from the two metering points meet in the collector. This means, that we can transmit time stamps of the two metering points to collector, and there we can make necessary calculations. Thus we minimize interventions to BasicMeter architecture and unnecessarily do not complicate it.

Exporter exports measured information about data flows: the flow identifiers, different time and volume characteristics, or information of the metering point. From the perspective of timestamps transfer, we will be interested in time characteristics exported from exporter to collector. Since the characteristics are transmitted from two different metering points, we need to combine related measurements in the collector. Therefore, each time characteristics needs appropriate flow identifiers.

If we manage to get in the collector time characteristics belonging to one packet flow, coming from two different metering points by comparing the flow identifiers, we can say that we managed to pass a time stamp from one metering point to another. Of course, we cannot take it literally, because to link and creation of ”transferred” timestamp became in the collector.

From IPFIX elements describing time characteristics we have chosen flowStartNanoseconds and flowEndNanoseconds as main. Some others could be used too, but with more effort. To be sure, that those two timestamps were given in different metering points from the time characteristic of the same packet, we need to identify those packets somehow. For this purpose we have created new information elements shown in Table I.

TABLE I
THE LIST OF CREATED IFPIX INFORMATION ELEMENTS FOR FIRST AND LAST IP PACKET IDENTIFICATION OF FLOW.

ID	Enterprise ID	Name	Type
242	26 235	firstPacketID	octetArray(16)
243	26 235	lastPacketID	octetArray(16)

For identifier generation, we have used a method from paper [9] about generation of unique packet identifier. With this method we can generate a 16 byte packet identifier from the selected fields of IP packet header and transmitted data of IP packet using MD5 hash function. Selected fields of IP packet header are as follows: IP header length, IP total length, Identification, Protocol, source IP address, destination IP address.

Method was tested on over 350 000 packets. There were 3.81% of non-unique identifiers. From this 3.81%, 2.64% were absolutely same packets, where have not been a chance to generate unique identifier. So in a fact just 1.17% was the mistake of the generator of identifier. [9]

C. One-Way Delay Calculation

Calculation of delay is expressed simply. Subtracting the two corresponding time stamps and the absolute value of this difference is the one way delay between two metering points. The corresponding time stamps are time stamps that meet the following conditions:

- both are either absolutely beginning of the flow times (flowStart) or absolute flow completion times (flowEnd)
- both have identical identifiers of IP packet
- both belongs to the same flow
- flow records came into the collector from different metering points

For storage of measured one-way delays were created new IPFIX information elements. They are shown in Table II. Each of them has assigned the enterprise ID of 26235. Observation point IDs start and end will contain identifiers of metering points. OWD elements will contain delay itself.

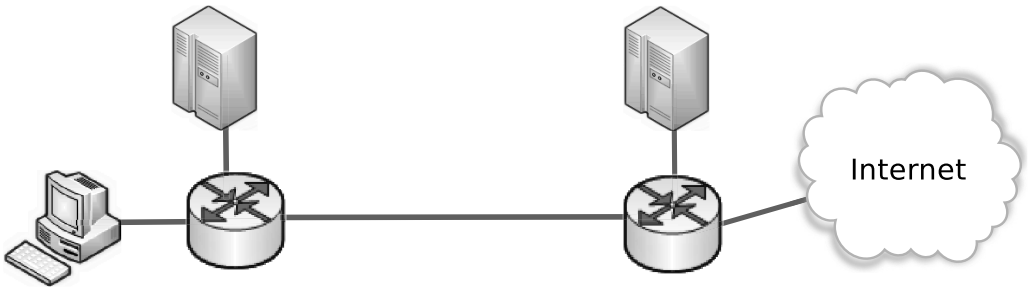


Fig. 2. Topology of the experimental network.

TABLE II
LIST OF CREATED IPFIX INFORMATION ELEMENTS FOR ONE-WAY DELAY STORAGE.

ID	Name	Type
244	owdStartObservationPointID	unsigned32
245	owdEndObservationPointID	unsigned32
246	owdSeconds	dateTimeSeconds
247	owdMiliseconds	dateTimeMiliseconds
248	owdMicroseconds	dateTimeMicroseconds
249	owdNanoseconds	dateTimeNanoseconds

For one flow, we can calculate the number of one-way delays between different pairs of metering points. It worth considering, to do not add the measured one-way delay in the flow record, but keep it separated, since the delay does not apply to the entire flow, but only to a path which flow passes through.

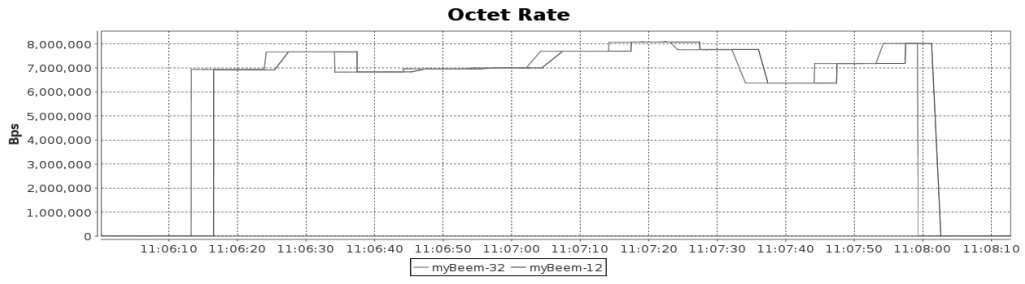
III. EXPERIMENTAL VERIFICATION OF PROPOSED SCHEME

The functionality of synchronization algorithm and method for measurement of one-way delay were experimentally verified on real network traffic. We have created a topology Fig. 2 consisting of two routers, two servers, personal computer and connectivity to the Internet.

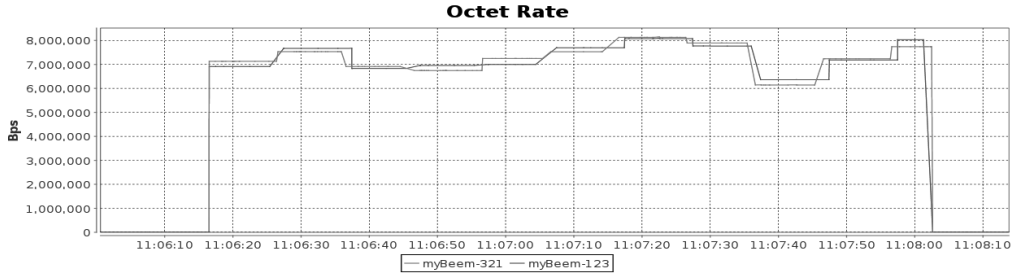
Each router is connected to a server. Server was presented as the metering point. Server was connected to the router's network interface configured as SPAN (switchport analyzer) of network interface directing towards the Internet. In this way we were able to capture network traffic between the PC and the Internet.

Server's system clocks were configured, so that the time difference between them was for approximately 3 seconds. This introduced the diversity of exported time stamps from metering processes.

Metering and exporting processes were four. Both servers were running two processes. One of them used a synchro-



(a) unsynchronized metering points



(b) synchronized metering points

Fig. 3. Octet rate measured for same flow on two metering points.

nization algorithm to synchronize against the collector. Every exporter used different observation point identifier. Collecting process was launched at one of the servers. Captured information were stored in a database. Data analysis was performed after the measurements.

Prepared topology allowed us to analyze the data flows passing through routers by two pairs of exporters. The first pair did not use synchronization and measured data in the database were inaccurate. Another pair did use synchronization.

On the PC we have started downloading a file from the Internet with the size of 700 megabytes. Created data flow has been captured, the information on it was saved to the database and the entire flow was analyzed. The resulting graphics of flow pass provided by BM analyzer are shown in Fig. 3a and Fig. 3b.

The first picture shows the measurements of pairs of exporters which are not synchronized. We see a delayed start and end of the flow between them. The second figure shows a pair of synchronized exporters. We see that the beginning and end of the flow is identical in this view. Note that in fact they are not identical. The difference between the start and end of a flow is one-way delay of connection between metering points. But this is so small that the difference in this representation is not noticeable.

Comparing the measured start and end time of the reference flow, we have got differences 149,248 and 82,944 nanoseconds, which are one-way delays of the flow between routers. Specifically, the delay of one Ethernet segment and the first router routing process.

We performed also experiments with the synchronization accuracy [10]. If high-speed links with a transmission speed of 100 Mbps are taken into consideration, the transmission of one ICMP packet (84 bytes) would take approximately 7 μs. This value is only theoretical, as in reality we also have to include the time for packet processing by the operating system of measurement points. The actual measured value is probably just one-half of RTT returned by the program ping (0.883 ms). In case of our experiments, the synchronization error was approximately 5 μs. This value is negligible (0.6 %) compared to the measured value. Without compensation, the clocks synchronization using public NTP servers is not accurate (1 ms accuracy of NTP [6] would cause 113 % error in this case of the measured time value).

IV. Conclusion

In this paper, we have presented problem of one-way delay measurement in the BasicMeter architecture. The BasicMeter represents an implementation of the IPFIX architecture, so this approach can be used in any implementation of IPFIX.

Along with the one-way delay measurement we have developed method for the time synchronization of metering points with compensation of the real time clocks' inaccuracy. The compensation method solves the problem of mutual inaccuracy of metering points clocks and, therefore, its application is required regardless of the synchronization method used, even in the case we have a high precision synchronization, e.g. using GPS signal.

Considering the measured inaccuracy of the real time clock, it is not desirable to adjust the real time clock frequently. On the contrary, this would lead to increased inaccuracy of the measured values, as the metering tool would not be able to distinguish the old timestamps from the new ones. In such case, the resulting value would be influenced by an error despite the usage of synchronization. More suitable approach is not to adjust the real time clocks, but compensate their mutual inaccuracy on the level of the metering tool. The disadvantage of this approach is that the calculation of one-way delay by half dividing of packet round trip time is not suitable in the case of an asymmetric link.

The future work of the authors will focus on the solution of the issues related to the synchronization convergence in extensive measuring networks with a large number of metering points. In addition, the problem of measuring link asymmetry will be a subject of further research by the authors. Also factors influencing instability of the inaccuracy rate will be researched, which is needed for determining of appropriate period for its recalculation.

Acknowledgment

This work is the result of the project implementation: Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

References

[1] O. Gurewitz, I. Cidon, M. Sidi: *One-way delay estimation using network-wide measurements*, IEEE/ACM Trans. Networking, pp. 2710-2724, June 2006

[2] P. Počta, P. Kortiš, P. Palúch, M. Vaculík: *Impact of the background traffic on speech quality in VoIP*, Measurement of speech, audio and video quality in networks: international conference and on-line workshop, Prague, 2007, CVUT, pp. 45-52, ISBN 978-80-01-03734-8.

[3] J. Smieško: *Link dimensioning with respect to QoS*, Journal of Information, Control and Management Systems, 2010, Vol. 8, No. 1, pp. 71-80, ISSN 1336-1716.

[4] M. Vrábel, I. Grellneth: *The "Amber" Approach to Active Queue Management*, In Proceedings of the 7th International Conference on Emerging e-Learning Technologies and Applications, ICETA 2009, November 19 - 20, 2009, Stará Lesná, High Tatras, Slovakia, ISBN 978-80-8086-128-5.

[5] S. B. Moon, P. Skelly, D. Towsley: *Estimation and Removal of Clock Skew from Network Delay Measurements*, Dept. of Comput. Sci., Massachusetts Univ., Amherst, MA, Marec 1999

[6] D. L. Mills: *Network time protocol (version 3): Specification, implementation and analysis*, RFC1305, 1992, pp. 112.

[7] D. L. Mills: *Modelling and analysis of computer network clocks*, Tech. Rep. 92-5-2, Electrical Engineering Department, University of Delaware, May 1992

[8] B. Claise, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information*, RFC 5101, January 2008.

[9] F. Jakab, G. Baldovský, J. Genči, J. Giertl: *Unique Packet Identifiers for Multipoint Monitoring of QoS Parameters*, Proceedings of ECI'2006, 7-th International Scientific Conf. on Electronic Computers and Informatics, Herľany, September 20-22, Herľany-Košice, FEEI TUKE, 2006, 6pp., ISBN 80-8073-150-0.

[10] M. Révés, J. Giertl, F. Jakab: *Improvement of Synchronization Accuracy in the Monitoring of Computer Networks*, The Mediterranean Journal of Computers and Networks, Vol. 4, No. 1, 2008, pp. 37-43, ISSN 1744-2397.

New approach to remote laboratory in regard to topology change and self–repair feature

Peter Feciľak, Katarína Kleinová
Dept. of Computers and Informatics, Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 04001 Košice, Slovakia
Email: {Peter.Fecilak,Katarina.Kleinova}@cnl.sk

***Abstract*—Remote laboratories deal with performing real lab experiments remotely via Internet. Recent advances in Internet/web technologies and computer-controlled instrumentation allow net–based techniques to be used for setting up remote real laboratory access. This paper deals with the solutions for remote laboratory providing services for the operation of remote laboratory. The paper addresses problems related to requirement of flexible, secure and easy remote access to laboratory equipment as well as the need for hardware and/or software re–configuration. This paper presents a unique concept for logical topology change with the usage of Q-in-Q tunneling and automated self–repair feature after failure.**

***Index Terms*—Virtual laboratory; Q-in-Q tunneling; AToM, Knowledge evaluation; Automated password-recovery; Logical topology; Remote access**

I. Introduction

Virtual laboratory in terms of this paper represents environment which is used for blended distance learning in Networking Academy program. Main purpose of virtual laboratory is to provide remote access to physical devices in network laboratory with the goal of reaching exercises on real devices placed in virtual laboratory environment as well as combining them with virtual devices and providing services for wide range of applications used in complex labs (like Authentication server, Virtual Private Network server, Active directory (domain) server, etc.).

Main areas which must be reflected by modern virtual laboratory are:

- Remote access to real or virtualized devices (defined interface)
- Devices maintenance (password-recovery procedure, power on/off)
- Reservation system
- Content for exercises (labs)
- Physical/Logical topology re-configuration (dependent on exercise)
- Knowledge evaluation system

This paper presents several drawbacks of existing solutions. Couple of solutions were already published in [2][4] and this paper follows with detailed solution for logical topology change/re–configuration and automated topology repair after failure. This paper also describes physical lab environment components which we have used in virtual laboratory at Academy Support Center at Košice.

II. Drawbacks of existing solutions

In this section we will pass through each area of modern virtual laboratory (chapter I) and describe some drawbacks of solutions that are used in virtual laboratories.

A. Remote access to real or virtualized devices

Depending on devices used in virtual laboratory, it is necessary to define an interface for remote access to equipment. In case of remote laboratory for computer networks we usually use network devices like routers and switches that can be managed over telnet/ssh protocol or by serial or auxiliary interface. In general, it is TCP/IP or serial communication interface (RS232) that is using 9600 bits per second speed by default.

The cheapest way to access devices remotely is by using their own TCP/IP interface for remote access like using telnet or ssh protocol. This solution has its weaknesses in the need of correct configuration of TCP/IP stack at used devices. In case that IP address will be re-configured by user of virtual laboratory, then device of virtual laboratory will be no more accessible remotely at defined IP address. Even in case that we will put some kind of warning such as ”do not re-configure interface, etc.”, we are unable to guarantee accessibility of virtual laboratory as it might be malfunctioned by user. Therefore it is more stable if remote access is based on terminal server that has defined interface for accessing the remote devices connected to terminal server. Usually it is telnet or ssh protocol used for remote access.

There are a lot of laboratories that are using terminal server with telnet access on different ports for each device connected to terminal server. Terminal server based on Cisco integrated services router (terminal server) with 8 to 32 serial asynchronous interfaces is mostly used. There is also need for possibility of direct access to terminal server settings, like clearing frozen sessions and changing port speeds. If there is no other user interface to communicate settings directly to terminal server, then there is no way to access devices with different speeds of console port than default. This can be lack of this solution. As soon as user will change the speed of console port during lab exercise configuration, device becomes unusable for next reserved sessions.

Direct access (even relayed through terminal server) to devices using telnet protocol might be problematic in networks

with too restrictive security policy. Due to security weakness of this protocol it is usually blocked in computer networks or service provider networks. Therefore there is strong need to provide secure way of communication with terminal server.

User interaction in remote network topology is also an important part of virtual laboratory. There is lack of virtual laboratories which combines remote laboratory equipment and user equipment with possibility of connecting own equipment (like user computer) into remote network topology. Usually remote network topology contains intermediate devices like routers and switches and there is lack of end devices like computers, IP phones and printers that can be controlled remotely. There is also strong need for terminal services not only for serial communication, but also for virtualization of operating systems and emulation of other network devices.

B. Devices maintenance

There are several actions that can be done by user and that can completely malfunction virtual laboratory. Therefore there is need for virtual laboratory equipment maintenance. These actions include:

- Re-configured passwords for console access or privileged exec mode *will cause inaccessibility of virtual laboratory for next users trying to access device*
- Changed speed of console or auxiliary port on device *will cause virtual lab device inaccessibility due to need for speed change at terminal server*
- Enabled security features that are blocking password-recovery procedure *will cause lab equipment to be unreachable due to impossibility of automatic password-recovery procedure*
- Erased flash memory *will cause device fails to boot and due to this problem it will not be accessible for lab training*

In modern virtual laboratories there is need for command authorization that cannot be done on Cisco ISR terminal server. Therefore a lot of virtual laboratories that are using Cisco terminal server are facing problems listed above and are solving them by person manually checking devices after each lab reservation. It is also possible to authorize commands on IOS application level with AAA server using tacacs or radius protocol. The solution using an authorization server has its weakness in that it relies on correct device configuration and its connectivity to AAA server. It is also limiting in case that authentication, authorization and accounting is part of exercise.

C. Reservation system

Each virtual laboratory has its own reservation system. Usually there is lack of easiness during reservation process. Some reservation systems are based on manual account creation (on devices or on terminal server) allowing user to access devices remotely. This process can be also partially or fully automated, which means that during reservation of lab session there is process including:

- Receiving of lab reservation request from community using virtual laboratory. There are different forms of request receiving - e-mail, web form, phone call to maintainer, etc.
- Approval and/or direct reservation
- Creating account and defining access rules
- Notification of person wishing to reserve lab equipment

Electronical requests (done via web form) can be almost fully automated, but there is also possibility of other non e-form requests to lab equipment maintainer. Therefore there is request for easy and fast process of equipment reservation integrated into traditional work user interfaces without spending too much time logging into reservation system, filling form items like e-mail of requester, date and time of lab reservation and notifying requester back.

D. Physical/Logical topology re-configuration

Sometimes it is necessary to re-configure network topology depending on the exercise that user wants to perform on virtual laboratory. There are a lot of virtual laboratories that do not allow topology re-configuration and all labs are based on the same topology or allow topology re-configuration only by technical staff physically changing network topology. There are also some approaches to automated change of physical topology based on connection matrices that physically interconnect wires by relay circuits. Some virtual laboratories are using logical topology change on ethernet network instead of physical topology re-configuration. There is issue for using solution based on VLANs for creating interconnection on L2 device for exercises related to L2 protocols like CDP, STP, VTP.

E. Content for exercises and knowledge evaluation

Every virtual laboratory has its technical limitations. Based on technical limitations there is limited set of exercises that can be done on set of equipment in virtual laboratory. Laboratories, that did not solve technically topological re-configuration, are usually dedicated to specific areas and therefore there is lack of scalability and possibility of doing wide range of exercises is typically missing. If laboratory is more static than dynamic in terms of topology creation, then there is usually no option for content creation (like connecting of devices together by web-oriented application such as packet tracer [1] application).

Important part of modern virtual laboratory is a system for knowledge evaluation. Based on exercise that is user doing in virtual laboratory there should be system for configuration collection and configuration files evaluation. There are number of virtual laboratories that are evaluating solution of exercise only by comparing solution file against user solution. Percentage of the difference between two solutions (template and user) is inverse percentage to 100%. There are still some problems with this solution as it is not so exact and also there is almost no variability in exercises (like IP address needs to be the same) and therefore exercise needs to be written so precisely that there is no other solution for the task.

III. Approach to topology change and self–repair feature

There are couple of solutions for drawbacks mentioned in chapter II which were already presented in [2] and [4]. In this section we will focus only on our unique approach to logical topology management with the usage of Q–in–Q [3] and automated password–recovery feature for routers and switches which is part of automated self–repair feature after failure.

A. Q–in–Q tunneling and its specific usage in remote laboratory with the goal of topology change

Q–in–Q (802.1q Tunneling) is mainly used by internet service providers offering the service for interconnection of customer sites on Layer2 ethernet technology. Goal of this technique is to double tag an ethernet frame and allow customer traffic separation. Figure 1 shows structure of double tagged ethernet frame.

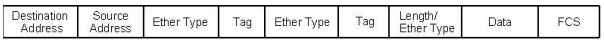


Fig. 1. Double tagged ethernet frame structure

802.1q enables service providers to use a single VLAN IDs to support customers who have multiple VLANs. On customer site, customer switch is connected to service provider switch and uses trunk mode on that port. On service provider switch is used 802.1q Tunnel mode. When Q-in-Q tunnelling is enabled, trunk interfaces are assumed to be part of the service provider network and access interfaces part of customer site. A trunk interface can be a member of multiple VLANs from service providers. An access interface can receive tagged or untagged frames. A customer has usually allocated only one VLAN from service provider and every frame coming from customer is tagged with allocated unique VLAN tag.

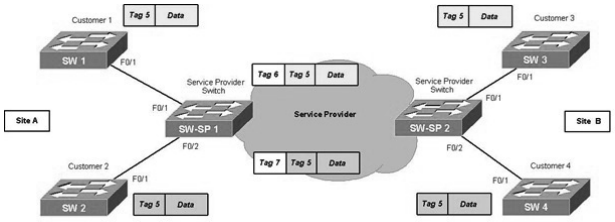


Fig. 2. Service provider network with Q–in–Q

Figure 2 shows service provider network when Q–in–Q double tagging is used for two customers. Every frame from customer on transmitted site is tagged at the service provider switch. The original VLAN tag of the frame is not changed and is transmitted transparently, so every frame has two VLAN tags (inner and outer tag) in service provider network. The inner tag is the customer VLAN tag, the outer tag is the service provider allocated VLAN tag. On receiving site, the outer tag is removed in the service provider switch and the frame with original customer tag is forwarded to customer. On the base of this, different customers can use the same VLAN tags.

Because of trunk mode on customer port, there are a few requirements:

- Port on service provider switch must set BPDU Filter and Root Guard to prevent customer switch to act as a STP root switch.
- It is necessary to configure protocol tunnelling to enable the VTP between customer switches (not working by default) in service provider switches.
- UDLD, PAgP and CDP (disabled by default) can work.
- VLAN ID field in the 802.1q frame is 12 bits therefore the maximum VLAN ID allocated to customer can be 4096.

There are few limitations for Q-in-Q tunnelling:

- Q-in-Q tunnelling does not support IGMP snooping or most access port security features.
- It is not available to disable MAC learning.
- There is no per-VLAN (customer) policing or per-VLAN (outgoing) shaping and limiting with Q-in-Q.

However, Q–in–Q tunneling provides great feature – the transparent interconnection of customers (physical ports of device). Our approach to logical topology building in virtual laboratory uses this feature.

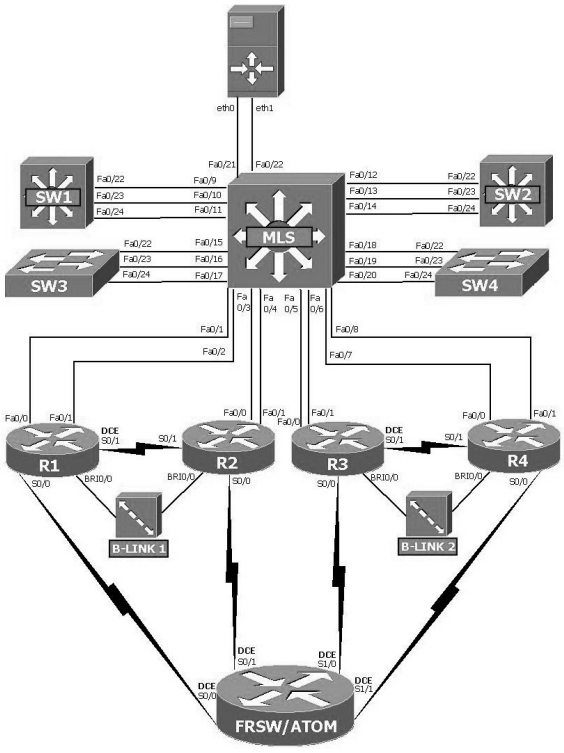


Fig. 3. Virtual laboratory topology

Our physical topology of virtual laboratory is shown on Figure 3. This network topology is physically static, we do not use any connection matrices as there is no need for doing

this. We have decided to manage topology more logically than physically by using Q–in–Q tunneling and Any Transport Over MPLS (AToM) [5]. These technologies allow us to logically create interconnections between each devices by using separated VLANs and to tunnel layer 2 protocols like CDP/DTP/STP by using Q–in–Q tunneling. Also interconnections between devices using serial interfaces (WIC-2T) can be done by frame relay circuits or by using encapsulation (tunneling) to MPLS (AToM). For each interconnection of devices we are using internally different VLAN to separate traffic and L2 tunneling techniques to provide transparent bridging of interfaces. Table I shows example configuration of virtual laboratory components when building logical topology from physical topology (Figure 3) shown on Figure 4.

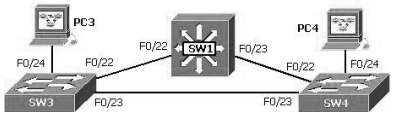


Fig. 4. Example of virtual laboratory topology

TABLE I
EXAMPLE OF MLS CONFIGURATION FOR INTERCONNECTION SW1-SW3

```
MLS(config)# interface range Fa0/9, Fa0/15
MLS(config-if-range)# switchport mode dot1-tunnel
MLS(config-if-range)# l2protocol-tunnel cdp
MLS(config-if-range)# l2protocol-tunnel stp
MLS(config-if-range)# l2protocol-tunnel vtp
MLS(config-if-range)# switchport access vlan 10
MLS(config-if-range)# description SW1-SW3
```

There is no need for hardware re-configuration in switching between exercises when Q–in–Q tunneling is used with the goal of establishing logical topology over the physical interconnections of laboratory. It is less expensive and easier to implement such feature of re-configuration as a part of software solution of virtual laboratory. Depending on exercise, this allows the user to load different configurations of logical topologies and to make full use of virtual laboratory resources.

B. Automated password-recovery feature for routers and switches

Automated password-recovery is a part of self–repair feature of the laboratory which is used for recovery of virtual/real equipment after failure due to user changes done during exercises. There are two different password-recovery actions that need to be supported by virtual laboratory. Password-recovery for routers is a bit different than for switches. There is strong need for speed change ability on each serial interface for successful password-recovery on router and mechanical push of MODE button for password-recovery on Catalyst switches. Key to password-recovery on routers is in control+break sequence generation. Practically this break sequence is generated by slowing speed down of console port lower than speed currently used and by sending 10 spaces. Therefore password-recovery on routers is done in following steps:

1) Power cycle the router (off/on)
2) Change speed of console port to 1200 bits per second
3) Send 10 spaces (0x20)
4) Change speed of console port back to default (9600 bits per second)
5) Configure config-register to 0x2142
6) Reload the router
7) Change config register back to default (0x2102)

Password-recovery procedure on Catalyst switches requires to manually push MODE button. The easiest way how to do this is by shorting MODE button circuit by contact relay managed from server. As we want to keep warranty on our virtual lab equipment, we have developed a unique prototype for manual pressing of MODE button. ”Buttoner” device is managed by SNMP and is mounted in rack on the top of catalyst switch. For the purpose of power control we have used SNMP managed power distribution unit (Figure 5).

Fig. 5. APC switched rack power distribution unit

C. Web user interface

Web user interface (Figure 6) acts as interface for communication with user. It is the central element for putting all the virtual laboratory pieces together. We have used some technologies like AJAX terminal that allows us to tunnel communication in case of limited access from user environment, PHP and Java technologies for running terminals from end station in non-firewalled environment.

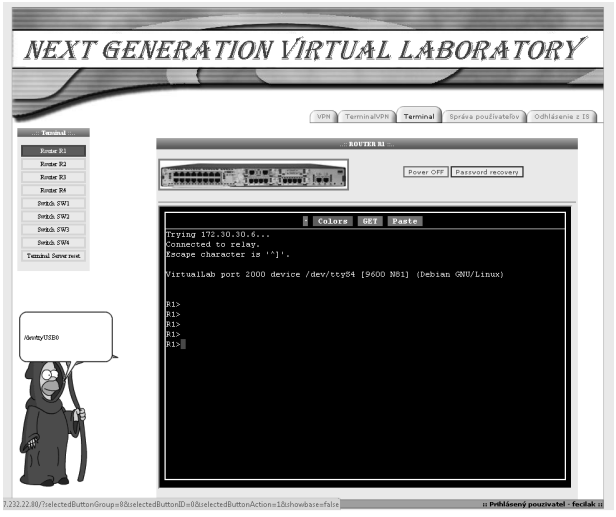


Fig. 6. Web user interface of virtual laboratory

IV. Conclusion

Remote laboratories have come a long way since their introduction in the late 1980s/early 1990s. The last decade witnessed a move from breaking technological barriers to the enhancement of pedagogical features, and the next generations will undoubtedly include embedded tutoring and personal assessment features that will improve their pedagogical effectiveness still further. The integration of remote experiments with the remaining e-learning contents offers an enormous potential for improving the pedagogical success of science and engineering students and therefore we will more focus on this topic in our future work.

In this paper we have presented several solutions for remote laboratory which we are operating at Academy Support Center at Košice. Main goal of this paper was to present our unique approach to topology change with the usage of service provider technology of Q–in–Q and to demonstrate our own approach to topology self–repair. This was clearly presented in chapter III. Our future work will be also devoted to videoconferencing as a part of training in virtual laboratory.

Acknowledgment

THIS WORK WAS SUPPORTED BY THE SLOVAK CULTURAL AND EDUCATIONAL GRANT AGENCY OF MINISTRY OF EDUCATION OF SLOVAK REPUBLIC (KEGA) UNDER THE CONTRACT NO. 3/7245/09(60%). THIS WORK IS ALSO THE RESULT OF THE PROJECT IMPLEMENTATION: DEVELOPMENT OF THE CENTER OF INFORMATION AND COMMUNICATION TECHNOLOGIES FOR KNOWLEDGE SYSTEMS (PROJECT NUMBER: 26220120030) SUPPORTED BY THE RESEARCH & DEVELOPMENT OPERATIONAL PROGRAM FUNDED BY THE ERDF (40%).

References

[1] Cisco Packet Tracer, [on-line 11.3.2011] URL: http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer.html

[2] Feciľak, P. – Kleinová, K. – Jakab, F.: *Solutions for virtual laboratory*, Proceedings of ICNS 2011, The Seventh International Conference on Networking and Services, May 22 to May 27 2011, Venice/Mestre, Italy, ISBN: 978-1-61208-133-5

[3] IEEE 802.1Q-in-Q VLAN Tag Termination, [on-line 11.3.2011] URL: http://www.cisco.com/en/US/docs/ios/lanswitch/configuration/guide/lsw_ieee_802.1q.html

[4] Jakab, F. – Janitor, J. – Nagy, M.: *Virtual Lab in a Distributed International Environment* - SVC EDINET, The Fifth International Conference ICNS 2009 on Networking and Services, LMPCNA 2009, Valencia, 20.-25. April 2009, Valencia, Spain, IEEE Computer Society, 2009, 5, ISBN 978-0-7695-3586-9

[5] Lobo, L. – Lakshman, U.: *MPLS Configuration on Cisco IOS Software*, Cisco Press 2006, ISBN: 978-1-58705-199-9

Simulation of Enhanced RADIUS Protocol in Colored Petri Nets

Jindrich JELINEK and Jiri FISER
Faculty of Science
J. E. Purkinje University in Usti n. L.
Usti nad Labem, Czech Republic
jindrich.jelinek@ujep.cz

Pavel Satrapa
Faculty of Mechatronics and Interdisciplinary Engineering Studies
Technical University of Liberec
Liberec, Czech Republic
pavel.satrapa@tul.cz

This article describes a model of the enhanced RADIUS protocol simulated in Colored Petri Nets. The model simulates the distributed network topology with one supplicant and several RADIUS servers and it is divided into some sections. One section is simple supplicant sending authentication requests. Next sections describe the RADIUS servers which evaluate the request and make a decision. Last section consists of the communication infrastructure. This model make possible solution of the problem with authentication a client if its home realm is not reachable and some else realms know the identity of this client. The model of protocol is realized in CPN Tools.

Petri Nets, CPN Tools, Computer Network, RADIUS

I. INTRODUCTION

This work deals with the problematic of distributed computer network with federated authentication. Common representation of these networks is networks based on RADIUS protocol [1], [4]. It is a rather simple protocol, which has currently many software implementations. For example, it is used by academic network *eduroam* [6]. RADIUS protocol in distributed environment transmits an authentication request of client to the home authentication server via an unreliable computer network. The home authentication server will accept or reject this request. One of disadvantages of current solution is possibility of failures if the home server is temporary unavailable. The system is currently set up as follows. If the home authentication server is unavailable or does not respond within a time limit, the server-recipient has to reject the request.

This article describes our model which can this problem minimalize and increase system reliability. The principle of the solution is based on the fact that a distributed network with federated authentication can be composed of many servers. The authentication information of the user can be stored not only in the home server, but also on other servers. The authentication information has been saved on other server when the user used this authentication server to log in to the network. This server stores the information and contacts the user's home server as described above. In the case of failure of the home authentication server our model will contact by a group message all servers in the domain and try to authenticate the user using their information.

II. INTRODUCTION TO PETRI NETS

Petri Nets was created in 1962 by Carl Adam Petri [5] and they are a mathematical representation of discrete distributed systems. The bases of Petri Nets are **places**, **transitions**, **arcs** and **tokens**. The tokens are located in the places and they are transmitted in the created network, depending on the status of places, transitions and arcs. The places contain a certain amount of tokens and transitions perform required operations with the tokens and the tokens are transferred through the arcs in the Petri Net. The arcs connect only the place and the transition, not two places or two transitions.

The Petri Nets were gradually developed over the years according to the needs of their users. Today, there are large numbers of different variants of Petri Nets. Petri Nets can be timed, stochastic and the so-called Colored Petri Nets (CPN) [3]. CPN is one of the high-level Petri Nets and allow modeling a wide range of problems. They can contain multiple types of tokens (e.g. colors) and these tokens can be treated differently. Tokens can also be numeric or string and so on. This allows you to model many aspects of network protocols and so on.

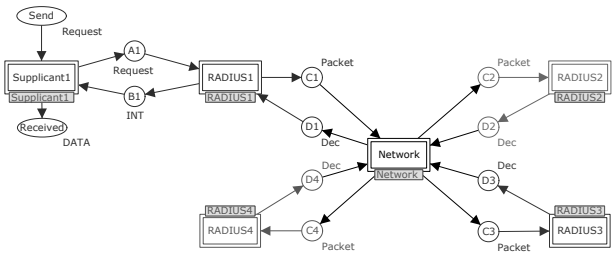


Figure 1. High Level Scheme of Model

III. MODEL OF THE PROTOCOL

Our model was created in Colored Petri Nets using CPN Tools [2]. It is a hierarchical model that has two levels. The high level is depicted in Figure 1. Here you can see six basic sections – *Supplicant1*, *RADIUS1-4* and *Network*. Each section will be described individually below. Places *Send* and *Received*

are part of the section *Supplicant1* and on the high level are positioned to control the activities of the model.

IV. SECTION SUPPLICANT1

This section has three parts. First part is responsible for sending of requests, the next part for their receiving. The place *N* is located in the second part and it is used to count the number of received packets. It can be used to limit the total number of the packets and thus stops the simulation. Final part is area around the transition *Gen*. This part is responsible for generate a random requests. The request consists of three parts (*Name*, *Realm* and *Pass*). The request is formed by user name, home realm name (for example *tul.cz*) and password. The figure shows data for random combinations which generator can generate. After the generator receives a reply on the previous request it generates a next request. See arc labeled *k*. Value of *k* is also controlled by a **guard** in the transition *Gen* (marked by square brackets near the transition *Gen*). This guard is formed by conditions *k <= 100*. If this condition is false, guard permanently disables the activity of the transition *Gen*. This will stop generating requests and thus stops the simulation.

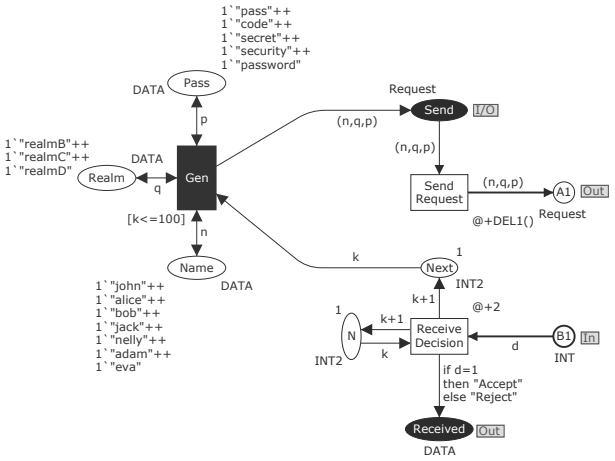


Figure 2. Scheme of Section Supplicant1

Some transitions and arcs in the model also include forms *@ + X*, where *X* is a time value. This value is added to the time value of the token when the token is passing through a transition or an arc and this token is delayed. This makes possible timing (e.g. delays, waiting etc.) in the model. Function DEL1(), respectively DEL2() used in some transitions adds a random delay to the token in the range 1-5, respectively 5-10 time units.

V. SECTION RADIUS1

This is the most complex section of the model; see Fig. 3 and Fig. 4. The area around the transition *Processing2* is responsible for processing requests, which directly targeted to server *RADIUS1* (i.e. requests containing the string "realmA"). In this configuration this type of requests (with "realmA") does not exist, because the generator in section *Supplicant1* does not generate them. This does not affect the results. All accepted authentication requests then continue into the auxiliary place *P* and then through the transition *Set* which is used to set the system to initial state. While requests are a triplets of values *(n, q, p)*, packets are formed by four values *(n, p, q, t)*.

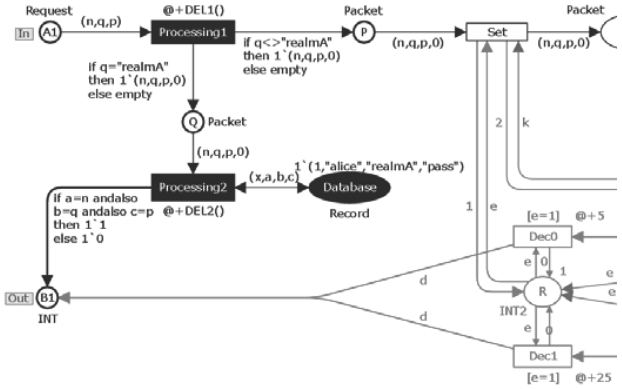


Figure 3. Scheme of Section RADIUS1 (Part One)

This section of the model was enhanced to improve the reliability of the system. The request which has not been processed by home server for a failure is going to be sent by server *RADIUS1* once again to all servers in the group (domain). Actual protocol RADIUS sends a request to an authentication only once. However other servers in the domain may use its database of previously stored authentication information. This information may include the authentication data of just authenticating user. This make possible user authentication. Current RADIUS servers do not have an auxiliary database, respectively do not use this.

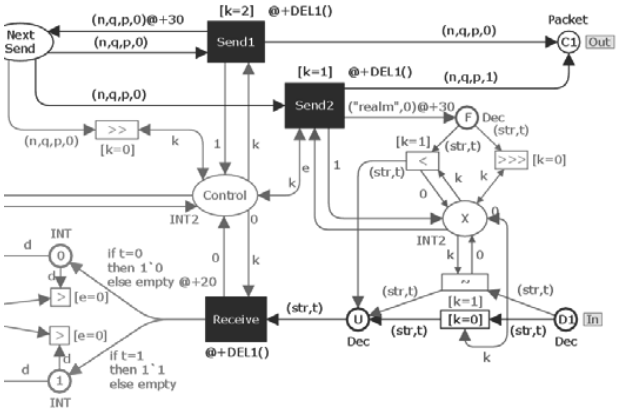


Figure 4. Scheme of Section RADIUS1 (Part Two)

Transition *Send1* sends packets that are targeted only to direct server – authenticated user's home server. Sent packet is composed of four values *(n, q, p, t)*, where *t = 0*. Value *t = 0* specifies that the packet will be delivered only to the user's home server. This packet is also stored in the place *Next Send* for possible later resend. Time value of stored packet is increased by 30 time units. During this time a response is received or a failure occurred (no response). Place *Control* is

also set so that resending of the packets does not go through the transition *Send1*, but through transition *Send2*. In the case that the authentication proceed and the answer is received, the answer will come to the place *D1* and passes through the transition marked *[k = 0]* and then through the transition *Receive* by one of paths to the place *B1*. If the answer passes through the transition *Receive* then the place *Control* is set fellows. The packet waiting in place *Next Send* is canceled (it pass through the terminal transition >> and disappear).

But if the home server response packet is not receive within the timeout, then the same packet is sent with *t = 1*. This packet is sent to all servers in the group and *RADIUS1* is waiting for their responses. But there is some probability that no response is received. In this case the server *RADIUS1* must respond itself by generating a formal negative response. To solve this situation the model use the subsection depicted around the place *X*. If the packet passed through transition *Send2* then a negative response is generated to the place *F*. There the response waits 30 time units and the state of the place *X* determine if the packet continue as the response (passing the transition <) or it is canceled (disappear the transition >>>).

Delivered response is finally forward to the section *Supplicant1*. After that, supplicant can send another request.

If the system receives more responses, it must make some decision (see the subsection around the place *R*, it can be called Adjudicator). There are two possibilities (positive or negative user's authentication). This subsection can be variously configured. In our case the model is set by time constants that negative responses are waiting in place *0*, while at least one positive response arrives. This response continues immediately (without delay) through the transition *DEC1*. Simultaneously the place *R* is set to delete all negative responses. Therefore the system prefers a positive authentication. Subsection can be altered in other ways. You can for example prefer a negative authentication or make qualified decision by some else adjudicator. It is possible to discuss different weight of response (e.g. user's home server). We assume that in the future, we will verify the different ways of decision-making model and the results will be presented.

VI. SECTION RADIUS2

Next sections of the model respond to the transmitted packets and make basics authentication. The model contains three almost identical sections *RADIUS2*, *RADIUS3* and *RADIUS4*. Sections differ only in particular records, which are stored in authentication databases. Sections *RADIUS3* and *RADIUS4* therefore will not be individually described.

Standard RADIUS server compares the information from the incoming packet with its database records. This function is modeled ~~the~~ subsection around the transition *Processing Local* and place *Database Local*. Auxiliary place *L* stores incoming information for comparison it with the corresponding record in the database. Each record (tuple *(x, a, b, c)*) is identified by the value of *x*. In place *N* is stored a value identifying the row which should be retrieved from the database. Comparisons and decisions are provided by arcs from the transition *Processing Local* to the place *D2*.

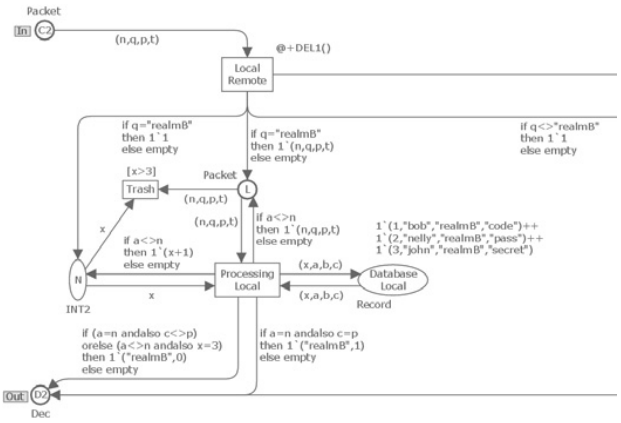


Figure 5. Scheme of Section RADIUS2 (Part One)

Enhanced RADIUS server contains a subsection around the transition *Processing Remote* and place *Database Remote*. It realizes a cache database that stores records of users, whose has been successfully authenticated to its home RADIUS server by a server *RADIUS2*. Place *Database Remote* contains records belonging to other realms than the realm server RADIUS2 (i.e. RealmB). The activities of this subsection are very similar to activities of subsection around transition *Processing Local*. The difference is that this area responds only to packets that have been sent repeatedly. This subsection also generates only a positive authentication results. If the user is unknown in *Database Remote* or if its authentication is negative, subsection does not generate any response. The reason is partly an effort to reduce the payload of the network and the fact that in the current configuration of *RADIUS1* prefers the positive results of authentication. Therefore negative results have no impact on result of system activity. In the case of change behavior of the server *RADIUS1* it is easy change the behavior of other servers to send a negative authentication too.

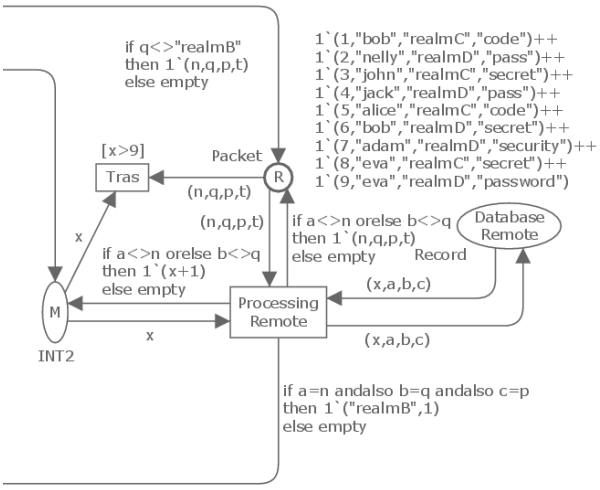


Figure 6. Scheme of Section RADIUS2 (Part Two)

VII. SECTION NETWORK

This section transmits packets from server *RADIUS1* to server from *RADIUS2* to *RADIUS4* and responding the decision from these servers to the server *RADIUS1*. The area between place *C1, C2*, *C3* and *C4* sends packets. If value *t* in the transmitted packet is set *t = 0* then the packet is transmitted only to the RADIUS server, which is determined by the value of the variable *q* (i.e. according to realm). If the value *t* in the transmitted packet is set *t = 1* then transition *Request2 Transmit* transmits this packet to the all servers.

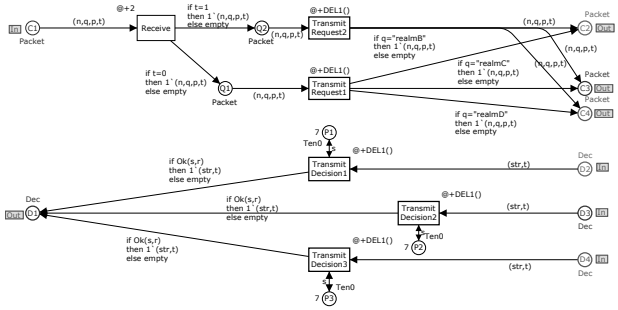


Figure 7. Scheme of Section Network

In the opposited direction transmission is realized by transitions *Transmit Decision1*, *Transmit Decision2* and *Transmit Decision3*. These transitions and adjacent arcs are equipped with the probability function *Ok (s, r)* [2]. This mechanism restricts the passing of a packet with a certain probability. The probability of successful passing is determined by value located in places from *P1* to *P3*. In our case the value is set to 7, i.e. the probability of passing is 70%. The estimated probability was chosen by guess. It is include all possible failures of servers and network that may occur. In the future we expect this probability will be set by real-world data.

VIII. CONCLUSION

The main goal of the model is to simulate an activity of the enhanced protocol RADIUS in a distributed environment. The model allows to verify the activity of improvements in various configurations and specific parameter settings (e. g. time constants, probability, etc.), as indicated above. It is easy to modify parameters of the model and configurations and validate the hypotheses about the behavior of computer networks, for example our model will be more successful for lower reliable networks.

Some experiments with parameters Authentication success rate and Duration of authentication are shown in the Fig. 8 and Fig. 9. These experiments were done for actual RADIUS protocol and for enhanced RADIUS protocol. Each model solved one hundred random authentication requests in a cycle (each referred value is the statistical average of ten cycles with specific network reliability). Our model with the probability set to 70% improves the ratio of positive authentication responses to nonlocal random requests from approximately 5,8% to approx. 10,2%. Time requirements increase only with coefficient approximately 1,29 (from 3900 to 5043 time units). It is evident from the graph in Fig 8., that for the lower reliability of network show the enhanced model better results.

The values of waiting for response (reaction time) were measured in experiments too. Response period in actual RADIUS protocol model is approx. 40,0 time units (reject) and approx. 37,2 time units (accept). Response period in enhanced RADIUS protocol model is approx. 48,4 time units (reject) and approx. 40,2 time units (accept). Referred values are the statistical average of five measuring with the network reliability set to 70 %. The values in time units will be converted into real time in future.

The probability of successful user authentication in presented solution is dependent on the presence of the user authentication information in other servers. In practice the probability of presence of authentication information in the servers is determined by frequency of the user's logins to various networks of the federation. Presented model has some input parameters, which can affect the probability of successful authentication (network reliability, number of servers, size of cache databases in servers, duration of cache records, number of combinations of input authentication requests etc.).

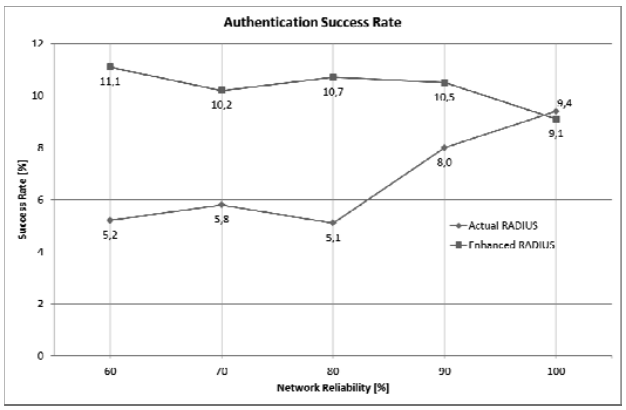


Figure 8. Graph of Authentication Succes Rate

The current configuration, however, differs from real network environment in adjustment of failure probability.

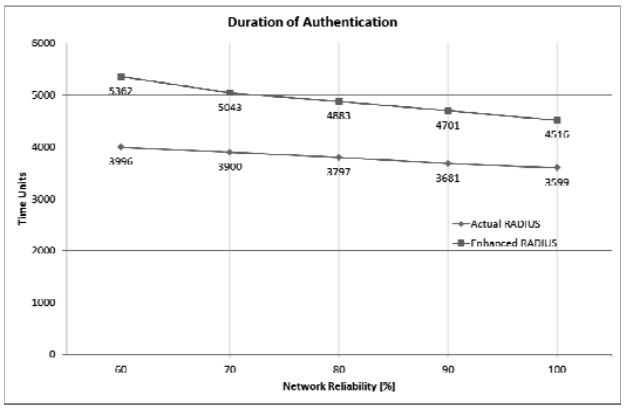


Figure 9. Graph of Duration of Authentication

Used value of the reliability (70 %) is not supported by a significant statistical data yet, but it is possible to see the reasons for some values of the network reliability in graphs. However, the number of authentication RADIUS servers is in the model very low (only three against a hundreds in the domain *.cz*) and databases of these servers contains fewer records than can be expected in real operation. Our model is going to be gradually improved to be more suitable for real operational deployments. The results can be used to implement in real protocols in distributed networks.

REFERENCES

[1] C. Rigney, Remote Authentication Dial In User Service (RADIUS), The Internet Society: Livingston, 2000, RFC2865.

[2] M. Westergaard, H. M. W. Verbeek, CPN Tools Homepage, CPN Tools, [Cited: 6. 4. 2011], http://cpntools.org/.

[3] K. Jensen, L. M. Kristensen, Colored Petri Nets, Springer-Verlag Berlin: Heidelberg, 2009.

[4] C. Rigney, RADIUS Accounting, The Internet Society: Livingston, 2000, RFC2866.

[5] C. A. Petri, Kommunikation mit Automaten, Ph.D. Thesis, University of Bonn, 1962.

[6] Author unknown, EduRoam Homepage, Terena, [Cited: 10. 6. 2011], http://www.eduroam.org/.

Specification, validation and verification of information transfer and processing systems

Ján Bača, Peter Feciľak
Dept. of Computers and Informatics, Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 04001 Košice, Slovakia
Email: {Jan.Baca,Peter.Fecilak}@tuke.sk

***Abstract*—Communication protocols represent part of transport environment in communication systems which are used nearly everywhere around us. Representing communication protocol or system itself by set of rules and conventions for communication is not enough for validation and verification of such system. Goal of this paper is to describe formal specification elements which allows us to realize validation and verification of communication system. We will focus on formal methods of describing communication systems with respect to requirement of exactness and completeness of specification. Objective of formalizing communication protocol or system is to get representation with further specification features.**

***Index Terms*—formal specification, validation, verification, protocol, information transfer**

I. Introduction

Systems for the delivery and processing of data represent complex distributed system. In the simplest case the system is built up with two separated workstations where the data exchange is happening over communication media. In more complex case there are number of stations interconnected and distributed all over the world which are faced to the problem of scalability and reachability.

From the point of view of networking service offered at the end station and/or server, there are different operations which are related to communication in data exchange process. Due to diverse nature of operations and data types which are being transmitted over the communication environment, basically the system is splitted into layers where each layer is responsible for accomplishing specific tasks.

One of the goals of this paper is to describe options for formal specification of protocols (covered in chapter III). Paper is also focused on validation and verification of protocols and/or systems (covered in chapter IV) and to the tools which were developed at the Department of Computers and Informatics, FEI, Technical university of Košice (see chapter V).

II. Network protocols

There are two models which are mainly used when explaining and defining operations in networking environment by layers. The ISO/OSI refference model and TCP/IP model which represents framework for computer network protocols and therefore it is known as protocol model. Communication protocol is a system of message formats and rules for exchanging those messages in or between computing systems. Operations at specific layers of ISO/OSI or TCP/IP model done by protocols are well defined and may have also formal description. In digital computing systems, the rules can be expressed by algorithms and data structures. Expressing the algorithms in a portable programming language, makes the protocol software operating system independent. The level of specification of protocol and/or communication system might differ depending on the tasks which are related to specific system. Any representation of protocol and/or data exchange system should clearly define:

- Data formats for data exchange
- Address formats for data exchange
- Address mapping and handling information
- Detection of transmission errors
- Flow control

The most widely-known TCP/IP Application layer protocols are those that provide for the exchange of user information. These protocols specify the format and control information necessary for many of the common Internet communication functions. Among these TCP/IP protocols are:

- Domain Name Service Protocol (DNS) is used to resolve Internet names to IP addresses
- Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web
- Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments
- Telnet, a terminal emulation protocol, is used to provide remote access to servers and networking devices
- File Transfer Protocol (FTP) is used for interactive file transfer between systems

The protocols in the TCP/IP suite are generally defined by Requests for Comments (RFCs). The Internet Engineering Task Force maintains the RFCs as the standards for the TCP/IP suite. In the OSI model, applications that interact directly with people are considered to be at the top of the stack,

as are the people themselves. Like all layers within the OSI model, the Application layer relies on the functions of the lower layers in order to complete the communication process. Within the Application layer, protocols specify what messages are exchanged between the source and destination hosts, the syntax of the control commands, the type and format of the data being transmitted, and the appropriate methods for error notification and recovery. Protocols establish consistent rules for exchanging data between applications and services loaded on the participating devices. Protocols specify how data inside the messages is structured and the types of messages that are sent between source and destination. These messages can be requests for services, acknowledgments, data messages, status messages, or error messages. Protocols also define message dialogues, ensuring that a message being sent is met by the expected response and the correct services are invoked when data transfer occurs.

Usually we are faced to the application layer protocols. However, the protocol is language with defined rules and basically also on the lower layers we have protocols which are used with the goal of assuring reachability, translation and security. For example TCP as the protocol is used for the reason of having connection oriented communication with automatic retransmition and error checking. If we are comming down to the bottom of the stack of ISO/OSI or TCP/IP model, we are faced to the protocols used for error detection and recovery on shared media (CSMA/CD, CSMA/CA), protocols used for address mapping like ARP as well as technology specific addressing protocols like Ethernet, DSL, ATM, Frame-relay, etc.

It is the strict need to have specification of protocol complete and integral to fulfill the needs for implementation. In some cases incomplete specification of protocol or system might cause that software engineer will be unable to implement the feature compatible with other specifications or might cause damage to the system where used (like frozing the system when input is not expected and exception was not caught). Common requirement for any protocol is the need for exactness and completeness of specification. Also it is very important to have options for specification validity checking. This is where computer system validation (CSV) takes place. CSV is the documented process of assuring that a computer system does exactly what it is designed to do in a consistent and reproducible manner.

III. Options for formal specification

Formal specifications represent techniques which are based on mathematical principles. Their usage allows to consistently check correctness of assignment, design and implementation of system. Inspection can be done automatically based on mathematical model of the system.

From the point of view of formalization it is possible to define protocol as language over the alphabet of events. Protocol strictly define which sequences of events are allowed.

There are couple of options for specification of protols, for example process algebra[1], petri nets[2], finite-state machines, specification languages like SDL (Specification and Description Language), LOTOS (Language Of Temporal Ordering Specifications), UML (Unified Modeling Language)[3], etc.

Formal specification differ from itself by options for text and graphical representation. Text form of representation is used for analysis of models and their exchange between tools. Graphical representation is used for the purpose of graphical visualization of model which might be represented firstly in text form.

IV. Validation and verification of specification

Basic requirements for analysis which allows us to validate defined system is check for exactness and completeness of specification. One of the possible ways of realizing correctness check is via finite-state machines. Mealy machine is defined as ordered 6-tuple (X, S, Y, σ, λ, S_0) where

- $X = \{X_1, X_2, ..., X_N\}$ – a finite (not empty) set of input states
- $S = \{S_1, S_2, ..., S_R\}$ – a finite (not empty) set of internal states
- $Y = \{Y_1, Y_2, ..., Y_M\}$ – a finite (not empty) set of output states

To simplify expression, in the following we will name input states simply as inputs, internal states as states and output states as outputs.

- $\sigma : SxX \rightarrow S$ – transition function – mapping pairs of a state and an input to the corresponding next state
- $\lambda : SxX \rightarrow Y$ – output function mapping pairs of a state and an input to the corresponding output
- S_0 – a start state (also called initial state) which is an element of (S)

There must be clear definition of transition for each state S_i and input X_j to make sure that definition of machine is correct. This also means that there are certain conditions which should be met:

1) No more than one transition should exist from any state S_i with the same conditions
2) Transition from any state S_i should exist

First condition represent **exactness condition** and second condition represent **completeness condition**. Both conditions must be true for every state $S_i \in S$ to make sure that machine is correctly defined.

For the reason of checking of conditions it is appropriate to express transition function in inverse form σ^{-1} where to every transition from state S_i to state S_j there is mapping of set of those input states where transition is entered. Inverse transition function represents the mapping

$$\sigma^{-1} : T \rightarrow 2^X \tag{1}$$

where T is set of all transitions and 2^X is set of all subsets of set of input states X. Input states are defined as combination of input variables x_k and can be expressed as

$$X_i : \prod_{k=1}^{n} \widetilde{x}_k^i \tag{2}$$

where $\widetilde{x}_k$ is equal to x_k or $\overline{x}_k$.

Set of input states $X_{i,j} = \{X_q | S_i x X_q \rightarrow S_j\}$ invoking transition from state S_i to state S_j defines transition condition which is marked as $L_{i,j}$. $L_{i,j}$ is defined as

$$L_{i,j} = \bigcup_{q} \prod_{k=1}^{n} \widetilde{x}_k^q \tag{3}$$

where $q \in \{1, 2, ...R | X_q \in X_{i,j}\}$.

Exactness condition is expressed by equation

$$L_{i,u}.L_{i,v} = 0, \forall i, u, v = 1, 2, ...R, u \neq v \tag{4}$$

Completeness condition is expressed by equation

$$\bigcup_{u=1}^{R} L_{i,u} = 1, \forall i = 1, 2, ...R \tag{5}$$

If transition from state S_i to state S_j does not exist, then $L_{i,j} = 0$. Therefore in completeness condition it is enough to consider only $L_{i,j} \neq 0$.

V. Application for verification of communication systems

When finite-state machines are used for representation of communication protocols and/or systems, even if they have many input and internal states, they usually have only couple of transitions. Also for this reason it is appropriate to define finite-state machine in form of inverse transition function. This approach was also used in in applications for verification of communication system which were described more detailed in [4][5]. Applications which were designed with the goal of verification of protocol have some little differences, however, their functions will be described together.

Very important part of application is its frontend for definition of finite-state machine which is done interactively. During the initial phase of defining finite-state machine user is allowed to continously enter its parameters which are mainly inputs, internal states and output actions of the system (outputs). For each parameter there is identifier and description of its meaning which facilitate the usage of the machine.

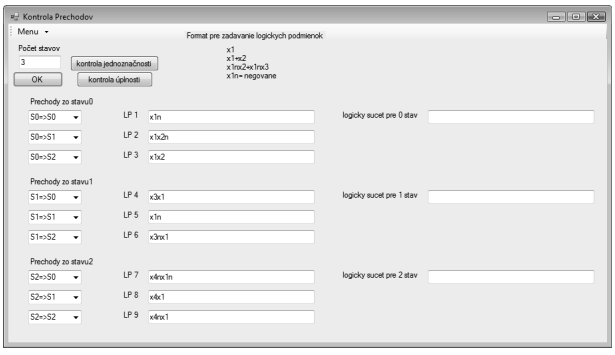


Fig. 1. Application user interface

In this application (fig. 1) which was developed as part of bachelor thesis in [4][5], inputs, states and outputs are defined first followed by entering particular transitions between states. It is possible to choose transition for any pair of created states. Subsequently it is possible to define logical condition in form of algebraic expression compiled from entered input variables and output actions which should be activated during transition.

After defining the finite-state machine, conditions check follows. This means that condition of completeness and exactness is checked. For each state, particular pair of logical conditions $L_{i,u}$ and $L_{i,v}$ is compared for $u \neq v$. If its penetration is detected then exactness condition is not met and user is warned by popup message. In this case logical conditions which are breaking exactness condition are marked with red. Completeness check is done after exactness check. If condition is not met then algebraic expression of sum of logical conditions for every transitions from defined state is shown. This might indicate that transitions were not properly defined – for example, some of transitions might be missing.

Application also supports saving and loading of defined finite-state machine into/from text-file representation.

VI. Conclusion

Application which was introduced in this paper (chapter V) represent easy way of specifying protocol via finite-state machine and allows to accomplish partial check of correctly entered specification. Also the protocol specification itself is hard to define. Sometimes also fulfillment of completeness and exactness conditions might be not enough. The only one specification which will met all requirements does not exist. Therefore we are usually using more specifications where each has its own advantages and disadvantages. It is very beneficial if the representation can be somehow transformed into other representation with the goal of gathering aditional properties of the system.

Process algebras are offering good modeling options like composition and decomposition of system from specification of individual agents. This represent very useful property of specification for communication protocols. Petri Nets are well known formal description language for its analytical properties of solving protocol verification problems, determining invariants, reachability, deadlock and liveliness[2].

For this reason we expect to expand application by options of entering protocols by the means of process algebra and its following transformation of specification to finite-state machine and Petri Net which allows us to do consistent verification of protocol specification.

Acknowledgment

THIS WORK IS THE RESULT OF THE PROJECT IMPLEMENTATION: DEVELOPMENT OF THE CENTER OF INFORMATION AND COMMUNICATION TECHNOLOGIES FOR KNOWLEDGE SYSTEMS (PROJECT NUMBER: 26220120030) SUPPORTED BY THE RESEARCH & DEVELOPMENT OPERATIONAL PROGRAM FUNDED BY THE ERDF (100%)

REFERENCES

[1] EDWARDS, J.: *Process Algebras for Protocol Validation and Analysis*, In Proceedings of PREP 2001, pages 1-20, Keele, England, 2001.

[2] ŠIMOŇÁK, S. – HUDÁK, Š. – KOREČKO, Š.: *Protocol Specification and Verification Using Process Algebra and Petri Nets*, Proceedings of CSSim 2009, First International Conference on Computational Intelligence, Modelling and Simulation, Brno, Czech Republic, 7-9 September 2009, pp. 110-114, ISBN 978-0-7695-3795-5.

[3] BABICH, F. – DEOTTO, L.: *Formal Methods for Specification and Analysis of Communication Protocols*, [online] 2002, [cit. 2011-08-03]. http://www.aleteya.cs.buap.mx/~jlavalle/papers/metodos-formales/FormalMethods.pdf

[4] EVIN, L.: *Formálne špecifikácie komunikačných systémov*. Bakalárska práca. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2011. 58s.

[5] FOLVARČÍK, P.: *Formálne prostriedky na špecifikáciu protokolov a validáciu ich korektnosti*. Bakalárska práca. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2011. 52s.

Composition of High Level Petri Nets

Ivan Peťko
Department of computers and informatics
Faculty of electrical engineering and informatics
Technical university in Košice, Slovakia
Email: ivan.petko@toryconsulting.sk

***Abstract*—Composition of High Level Petri Nets in terms of joining relevant places and/or transitions is considered in the paper. The approach is general with respect to the principles used for Low Level Petri Nets. In case of place composition type safe and combined types composition is contemplated. A formal background is established and some properties analysed in an informal way. The process of composition is proposed and analysed in three separate cases with respect to the general approaches with minimal composition interface in each case but an analogous extension of the interface follows immediately from the approach introduced.**

I. Introduction

Formal description techniques (FDT) are widely regarded as the only tool with ability to design, analyse and maintain complex discrete systems used in real word applications. Several FDT have been proposed in this field of academical research, from which probably one of the best known one are Petri nets. These provide very simple designing tool and they are appreciated especially for their simplicity and analytical properties.

Petri nets (PN) have been developed from the first proposal by C.A. Petri [7] and a wide family have been discovered covering many aspects of real word systems and even including advantages of some other FDT, for instance stochastic/time extensions [6], object Petri nets [9], algebraic PN [11], [14] and so on. The most significant extension in general are High Level Petri Nets (HLPN) [5]. The extension was proposed for several types of Petri nets including time aspects. HLPN provide very high modelling power altough their analysis is very difficult.

Since the first PN proposal one of the main reservations is the one about their unability of de/composition which is actually not included in the original conception. This motivated a lot of research and several de/compositional approaches (e.g. [2], [4], [3], [13]) including separate classes of de/compositional PN [8] have been proposed for modelling and/or analysis of Petri nets.

In the paper we focus on composition of HLPN. Instead of defining a separate class of composable HLPN or defining composition by means of compositional operators similar to the ones used in process algebras which have been proposed early on [2], [4], we concentrate on the HLPN class defined in the international standard [5] and composition is carried out as joining relevant places and/or transitions forming the interface of composition. In the first section the HLPN definition is introduced. Subsequently the process of composition is considered in three separate cases - place composition, transition composition and place-transition composition. The cases are considered with minimal composition interface but an analogous extension to more elements in the interface follows immediatly from the definitions introduced.

II. HLPN definition

In order to investigate composition of HLPN we focus on the HLPN standard [5]. The authors of the standard claim that it covers the ideas forming basic HLPN classes, namely Pr/T nets [12], colored nets [1] and algebraic nets [10]. There are some preliminaries we leave out in this paper such as multiset or formal term definitions. For more detailed information we refer to [5]. The standard includes two main definitions – HLPN and HLPN graphs. Since composition is more illustrative in the case of HLPN graphs, we consider them a base for our treatment and refer to this class as $HLPN(G)$.

Def. 1: HLPN graph ($HLPNG$) is a structure

$$HLPNG = (NG,\ Sig,\ V,\ H,\ Type,\ AN,\ m_0),$$

where

- $NG = (P,\ T,\ F)$ is a net graph with
  - P – set of places
  - T – set of transitions
  - $F \subseteq (P \times T) \cup (T \times P)$ – set of directed arcs referred to as flow relation
- $Sig = (S,\ O)$ is a many-sorted Boolean signature with the set of sorts $S = \{Boolean\}$ and boolean operations $O = \{<,\ \leq,\ \neq,\ =,\ ...\}$
- V – S-indexed set of variables, $V \cap O = \emptyset$
- $H = (S_H,\ O_H)$ - many-sorted algebra for the signature Sig defining its meaning
- $Type:\ P \to S_H$ – mapping assigning types (sorts) for places
- $AN = (A,\ TC)$ – net annotation with
  - $A:\ F \to Term(O \cup V)$ - mapping assigning terms to arcs. The result of term evaluation is a multiset over the types of associated places, i.e. $\forall((p,\ t),\ (t',\ p) \in F)\ \forall\alpha : Val_\alpha(A(p,\ t)),\ Val_\alpha(A(t',\ p)) \in Bag(Type(p))$, where $Term(O \cup V)$ is a set of terms over variables and operations, α is an assigning of token values to variables, $Val_\alpha(term)$ is term evaluation and $Bag(B)$ is a set of multisets over B

$TC:\ T \to Term(O \cup V)_{Bool}$ is a mapping assigning boolean expressions to transitions

$m_0:\ P \to \bigcup_{p \in P} Bag\,(Type\,(p))$ is initial marking

Def. 2: Marking of $HLPNG = (NG,\ Sig,\ V,\ H,\ Type,\ AN,\ m_0)$ is a mapping

$$m:\ P \to \bigcup_{p \in P} Bag\,(Type\,(p)),$$

such that $\forall p \in P:\ m(p) \in Bag\,(Type\,(p))$.

III. HLPN COMPOSITION

HLPN composition in contrast to the low level one has to take into account the net notations, i.e. a set of arc expressions and transition conditions. Moreover, since HLPN from definition contain a number of types for particular places, composition must take into account the types of these places. Composition of HLPN viewed as bipartite graphs may be performed through a set of places P_c, transitions T_c or both. We call the places P_c and transitions T_c the interface of composition denoted as I_c. In terms of [13] we have P composition if $I_c = P_c$, T composition if $I_c = T_c$ and PT composition if $I_c = P_c \cup T_c$. Note that we consider composition a junction. Composition may be divided into the following two steps:

1) structural composition
2) composition of net annotation

In the following we consider particular composition approaches as inverse operations to decomposition and define the resulting net as a compound of two subnets. It is clear that composition may be generalized for n subnets. We focus on the elementary cases (in case of P composition $I_c = \{p\}$, T composition $I_c = \{t\}$ and for PT we have $I_c = \{p,\ t\}$) but an analogous extension of I_c to more elements is possible provided that the composition definition is extended properly.

Let $N_1,\ N_2 \in HLPN(G),\ N_i = (NG_i,\ Sig_i,\ V_i,\ H_i,\ Type_i,\ AN_i,\ m_{0i}),\ i = \{1,\ 2\}$, and χ_C be a function defined on the $HLPN(G)$ domain such that

$$\chi_C:\ HLPN(G) \times HLPN(G) \to HLPN(G),$$

where $C \in \{P,\ T,\ PT\}$ provided that the signatures $Sig_1 = (S_1,\ O_1)$ and $Sig_2 = (S_2,\ O_2)$ do not contain the same operator definitions using different number nor different types of arguments, i.e. the following hold

$$\forall op \in O_1 \cup O_2:\ op_{(\sigma_1,\ s_1)} \in O_1 \wedge op_{(\sigma_2,\ s_2)} \in O_2 \Longrightarrow$$

$$\sigma_1 = \sigma_2 \wedge s_1 = s_2,$$

where op is the same operator defined in the signatures.

In order to define composition correctly let us introduce an auxiliary function

$$ext_{A,B}:\ Bag(A) \to Bag(B),$$

provided that $A \subseteq B$. The function represents an extension of the multiset over A to the multiset over B such that

$$\forall b \in bag\,(B):\ bag\,(b) = bag\,(b) \Leftrightarrow b \in A,$$

$$bag(b) = 0 \Leftrightarrow b \notin A,$$

where $bag(x)$ stands for multiplicity of the x element in the relevant multiset and $bag(B) \in Bag(B)$.

A. P composition

P composition is carried out as an inverse operation to P decomposition, i.e. a junction of places. Let $p_1 \in P_1$ and $p_2 \in P_2$ and $Type_1(p_1) \subseteq Type_2(p_2) \vee Type_2(p_2) \subseteq Type_1(p_1)$. Then $N = \chi_P(N_1,\ N_2)$ we call type safe composition. If $\exists(c_1 \in Type_1(p_1),\ c_2 \in Type_2(p_2)):\ c_1 \neq c_2$ then composition is of combined place types. Such composition is different from the type safe composition mechanism, i.e. it is possible in principle but it may not keep some of net properties after composition, e.g. boundedness. On the other hand type safe composition preserve boundedness if the both places to be joined are bounded.

a) P type safe composition: The resulting net $N = \chi_P(N_1,\ N_2)$ of composition of two subnets through places p_1, p_2 is given as

$$N = (NG,\ Sig,\ V,\ H,\ Type,\ AN,\ m_0),$$

where

$NG = (P,\ T,\ F)$
$P = P_1 \cup P_2 \cup \{p\} - \{p_1,\ p_2\}$
$T = T_1 \cup T_2$
$F = F_1 \cup F_2 \cup \{(p,\ t`),\ (t`,\ p) | t` \in T$
$\wedge (p_i,\ t`),\ (t`,\ p_i) \in F_i \Rightarrow (p,\ t`),\ (t`,\ p) \in F\}$
$-\{(p_i,\ t`),\ (t`,\ p_i) |\ t \in T\},\quad i \in \{1,\ 2\}$
$Sig = Sig_1 \cup Sig_2 = (S_1 \cup S_2,\ O_1 \cup O_2)$
$V = V_1 \cup V_2$
$H = H_1 \cup H_2 = (S_{H_1} \cup S_{H_2},\ O_{H_1} \cup O_{H_2})$
$Type:\ P \to (S_{H_1} \cup S_{H_2}),\ \forall p' \in P_i \backslash \{p_1,\ p_2\}:$
$Type(p') = Type_i(p'), Type(p) = Type_1(p_1),$
if $Type_2(p_2) \subseteq Type_1(p_1), Type(p) = Type_2(p_2),$
if $Type_1\,(p_1) \subseteq Type_2\,(p_2)$.
The latter may be written in general as
$Type(p) = Type(p_1) \cup Type(p_2)$
$AN = (A,\ TC)$,
$A:\ F \to Term\,(O_1 \cup O_2 \cup V)$,
$A(f) = A_i(f)$ if $f \in F_i \backslash \{(p_i,\ t`),\ (t`,\ p_i) | t` \in T\},$
$A\,((p,\ t`)) = A\,((p_i,\ t`))$,
$A\,((t`,\ p)) = A\,((t`,\ p_i))$,
for $(p_i,\ t`)$ , $(t`,\ p_i) \in F_i,\ t` \in T$
$TC:\ T \to Term(O_1 \cup O_2 \cup V)_{Bool},$
$\forall t` \in T : TC(t`) = TC_i(t`),$
$m_0 = ext_{P_1 - \{p_1\},\ P}(m_{01}|_{p_1}) + ext_{P_2 - \{p_2\},\ P}(m_{02}|_{p_2}),$
i.e. sum of initial markings of the subsets $N_1,\ N_2$ over P,
where $m_{0i}|_{p_i}$ is a restriction of the marking m_{0i}
to all places except $p_i,\ m_0(p) = m_0(p_1) + m_0(p_2)$

Notice that composition extends the state space of the resulting net to composition of state spaces of the subnets.

b) P combined types composition: It is possible to compose subnets through places of different types and the result is similar as in the case of P type safe composition with the following difference

$$Type:\ P \rightarrow (S_{H_1} \cup S_{H_2}),\ Type(p) = Type(p_1) \cup Type(p_2)$$

Composition of different types extends the marking of p so that the type of p is union of types of p_1 and p_2. On the other hand P type safe composition does not extend the type of p.

Both P compositions as introduced do not preserve boundedness if there is "insufficient" token consumption from p which is a junction of p_1 and p_2. The tokens of p may be removed if there are outgoing arcs from p assigning the tokens of the both types of p_1 and p_2 (each arc the respective type or at least one arc the both types) in the first or the second subnet being composed. Therefore if p_1 or p_2 is not bounded nor p is. Simultaneously if p_1 is not bounded in N_1 (p_2 in N_2) p may be bounded after composition provided that there is an arc in N_2 from p_2 (in N_1 from p_1) able to assign the tokens of the both types or more arcs each for the respective type.

B. T composition

T composition is carried out as an inverse operation to T decomposition, i.e. a junction of transitions. Let $t_1 \in T_1,\ t_2 \in T_2$ and t be the resulting transition. The transition condition for t is a logical disjunction of the conditions of t_1 and t_2.

The resulting net $N = \chi_T(N_1,\ N_2)$ of composition of two subnets through transitions $t_1,\ t_2$is given as

$$N\ =\ (NG,\ Sig,\ V,\ H,\ Type,\ AN,\ m_0)\,,$$

where

$NG = (P,\ T,\ F)$,
$P = P_1 \cup P_2$
$T = T_1 \cup T_2 \cup \{t\} - \{t_1,\ t_2\}$
$F =\ F_1 \cup F_1 \cup \{(p\text{`},\ t),\ (t,\ p\text{`})|p\text{`} \in P$
$\wedge(p\text{`},\ t_i),\ (t_i,\ p\text{`}) \cup F_i \Rightarrow\ (p\text{`},\ t),\ (t,\ p\text{`}) \in F\}$
$-\{(p\text{`},\ t_i),\ (t_i,\ p\text{`})|\ p\text{`} \in P\}$,
$Sig = Sig_1 \cup Sig_2 = (S_1 \cup S_2,\ O_1 \cup O_2)$
$V = V_1 \cup V_2$
$H = H_1 \cup H_2 = (S_{H_1} \cup S_{H_2},\ O_{H_1} \cup O_{H_2})$
$Type:\ P \rightarrow (S_{H_1} \cup S_{H_2}),\ Type = Type_1 \cup Type_2,$
$\forall p' \in P_i:\ \ Type(p') = Type_i(p')$
$AN\ =\ (A,\ TC)\,,$
$A:\ F \rightarrow Term(O_1 \cup O_2 \cup V),$
$A(f) = A_i(f)$ if $f \in F_i \backslash \{(p',\ t_i),\ (t_i,\ p')|p' \in P\}$,
$A\,((p',\ t)) = A\,((p',\ t_i))\,,$
$A\,((t,\ p')) = A\,((t_i,\ p'))\,,$
for $(p',\ t_i)\,,\ (t_i,\ p') \in F_i,\ \ p' \in P,$
$TC:\ T \rightarrow Term\,(O_1 \cup O_2 \cup V)_{Bool},$
$\forall t\text{`} \in\ T_i \backslash \{t_i\}:\ TC\,(t\text{`}) = TC_i\,(t\text{`})\,,$
$TC(t) = TC_1(t_1) \vee TC_2(t_2)$, i.e. the transition condition of t is logical disjunction of the original transition conditions of t_1 and t_2
$m_0 =\ ext_{P_1,P}(m_{01})\ +\ ext_{P_2,P}(m_{02})$, i.e. sum of initial markings of the subsets $N_1,\ N_2$ over P,
$m_0 \in Bag(P)$

Notice that T composition extends the state space of the resulting net N to composition of the state spaces of the subnets. Furthermore it does not change the place types as well as P type safe composition. The choice of logical disjunction of transition conditions is justified by the fact that logical conjunction would block the firing of the transition (if $TC(t_1) = \neg TC(t_2) \Rightarrow TC(t_1) \wedge TC(t_2)\ =\ false$). Disjunction is therefore a less strict condition and allows the execution in N in a more favorable way.

C. PT composition

PT composition combines features of P and T composition so it is based on joining places and transitions at the same time as an inverse operation to PT decomposition. Consequently P and T composition are special cases of more general PT composition. In order to point out the principle let us consider the easiest case – composition through places $p_1 \in P_1,\ p_2 \in P_2$with the resulting place p and transitions $t_1 \in T_1,\ t_2 \in T_2$with the resulting transition t.

The resulting net $N = \chi_{PT}(N_1,\ N_2)$ is given as

$$N = (NG,\ Sig,\ V,\ H,\ Type,\ AN,\ m_0),$$

where

$NG = (P,\ T,\ F)$,
$P = P_1 \cup P_2 \cup \{p\} - \{p_1,\ p_2\}$
$T = T_1 \cup T_2 \cup \{t\} - \{t_1,\ t_2\}$
$F =\ F_1 \cup F_2 \cup \{(p',\ t),\ (t,\ p')|p' \in P$
$\wedge(p',\ t_i),\ (t_i,\ p') \in F_i \Rightarrow (p',\ t),\ (t,\ p') \in F\}$
$\cup\{(p,\ t'),\ (t',\ p)|\ t' \in T$
$\wedge(p_i,\ t'),\ (t',\ p_i) \in F_i \Rightarrow (p,\ t'),\ (t',\ p) \in F\}$
$\cup\{(p,\ t)\,,\ (t,\ p)|\,(p_i,\ t_i)\,,\ (t_i,\ p_i) \in F_i$
$\Rightarrow (p,\ t),\ (t,\ p) \in F\} - \{(p_i,\ t_i),\ (t_i,\ p_i),$
$(p_i,\ t'),\ (t',\ p_i),\ (p',\ t_i),\ (t_i,\ p')|$
$t' \in T_1 \cup T_2,\ p' \in P_1 \cup P_2\}$,
$Sig = Sig_1 \cup Sig_2 = (S_1 \cup S_2,\ O_1 \cup O_2)$
$V = V_1 \cup V_2$
$H = H_1 \cup H_2 = (S_{H_1} \cup S_{H_2},\ O_{H_1} \cup O_{H_2})$
$Type:\ P \rightarrow (S_{H_1} \cup S_{H_2}),\ Type = Type_1 \cup Type_2,$
$Type(p) = Type(p_1) \cup Type(p_2)$
$AN = (A,\ TC)\,,$
$A:\ F \rightarrow Term(O_1 \cup O_2 \cup V),$
$A(f) = A_i(f)$
if $f \in F_i \backslash \{(p_i,\ t'),\ (t',\ p_i),\ (p',\ t_i),\ (t_i,\ p'),$
$(p_i,\ t_i),\ (t_i,\ p)|t\text{`} \in T_1 \cup T_2,\ p\text{`} \in P_1 \cup P_2\}$,
$A((p',\ t)) = A((p',\ t_i))$,
$A((t,\ p')) = A((t_i,\ p'))$,
$A((p,\ t')) = A((p_i,\ t'))$,
$A((t',\ p_i)) = A((t',\ p))$,
for $(p_i,\ t')\,,\ (t',\ p_i)\,,\ (p',\ t_i)\,,\ \ (t_i,\ p') \in F_i,$
$t' \in T_1 \cup T_2,\ p' \in P_1 \cup P_2$
$A\,((p,\ t)) = A\,((p_1,\ t_1)) + A\,((p_2,\ t_2))\,,$
$A\,((t,\ p)) = A\,((t_1,\ p_1)) + A\,((t_2,\ p_2))\,,$
where $A(f)=\emptyset \Leftrightarrow f \notin F$,
$TC:\ T \rightarrow Term\,(O_1 \cup O_2 \cup V)_{Bool},$

$\forall t' \in T_i \backslash \{t_i\} : \; TC\,(t') = TC_i\,(t')\,,$
$TC(t) = TC_1(t_1) \vee TC_2(t_2)$
$m_0 = ext_{P_1-\{p_1\},P}(m_{01}|_{p_1}) + ext_{P_2-\{p_2\},\;P}(m_{02}|_{p_2})$,
$m_0\,(p) = m_{01}\,(p_1) + m_{02}\,(p_2)$, i.e. sum of initial markings of the subsets $N_1,\; N_2$ over P, where $m_{0i}|_{p_i}$ is a restriction of the marking m_{0i} to all places except $p_i,\; m_0 \in Bag(P)$

Properties of PT composition are union of properties of P and T composition as outlined above.

IV. Conclusion

In this paper composition of the chosen HLPN class is considered. The class covers three main HLPN classes - predicate/transition nets, colored nets and algebraic nets and it is established in the international standard. The approach used comes out from general principles of composition used for low level PN in terms of joining relevant places and/or transitions. Another approaches are possible such as composition using special compositional operators but the aim of the paper is to introduce composition for the chosen HLPN class without any extensions needed.

Since (HL)PN are bipartite graphs in their graphical form, composition in our approach respects the low level principles and is defined as place, transition or place-transition junction. The places/transitions form the interface of composition and we considered only the minimal interface in each case believing that the respective extension to more elements is clear enough from the sketched approach. In the case of place composition type safe and composition of places of different types are considered separately because of the natural feature of HLPN - types of places. Type safe composition combines places of the same type or may be used when the type of one place is a subtype of the second. Composition of different place types brings some important corollaries especially in terms of keeping some of the net properties such as boundedness. Place and transition compositions are special cases of more general place-transition composition.

References

[1] K. Jensen, "Coloured Petri Nets: A high-level Language for System Design and Analysis", LNCS vol. 483, Springer Verlag, 1990

[2] I. Rojas, "Compositional construction and analysis of Petri net systems", Dissertation, University of Edinburgh, Edinburgh, 1997. Available: http://www.era.lib.ed.ac.uk/handle/1842/421

[3] J. Esparza, M. Silva, "Compositional Synthesis of Live and Bounded Free Choice Nets", Universidad de Zaragoza (Spain), Dpto. Ingeniera Electrica e Informatica, Grupo Ing. de Sistemas Informatica, Technical Report Mar, 1990

[4] E. Best, A. Lavrov, "Generalized Composition Operations on High-Level Petri Nets", *Fundamenta Informaticae 34*, pp. 1-39, IOS Press, 2000, Available: http://parsys.informatik.uni-oldenburg.de/ best/publications/best-lavrov-composop.ps

[5] The ISO website [online]. ISO/IEC 15909-1:2004, High-level Petri Nets - Concepts, Definitions and Graphical Notation, Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=38225

[6] B. Gianfranco, "Introduction to stochastic Petri nets", Springer Lectures On Formal Methods And Performance Analysis, Lectures on formal methods and performance analysis: *first EEF/Euro summer school on trends in computer science*, pp. 84-155, 2002, ISBN:3-540-42479-2

[7] C.A. Petri, "Kommunikation mit Automaten", Bonn: Institut fr Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962, Second Edition:, New York: Griffiss Air Force Base, Technical Report RADC-TR-65–377, Vol.1, 1966, pp: Suppl. 1, English translation

[8] H. Klaudel, F. Pommereau, "M-nets: a survey", *Acta Informatica*, Vol. 45, numbers 7-8, Springer Berlin / Heidelberg, pp. 537-564, 2008

[9] E. Battiston, F. DeCindio, G. Mauri, "OBJSA nets: a class of high-level nets having objects as domains", Springer Lecture Notes In Computer Science, Advances in Petri Nets 1988, pp. 20-43, ISBN:3-540-50580-6, 1988

[10] J. Vautherin, "Parallel systems specifications with Coloured Petri nets and algebraic specifications", *European Workshop on Applications and Theory of Petri Nets*, pp. 293–308, 1986

[11] E. Best, R. Devillers, M. Koutny, *Petri net algebra*, XI, 378 p., Hardcover, ISBN: 978-3-540-67398-9, 2001

[12] H.J. Genrich, "Predicate/transition nets", *Advances in Petri nets 1986*, part I on Petri nets: central models and their properties, pp. 207 — 247, ISBN:0-387-17905-4, Springer-Verlag, 1987

[13] Š. Hudák, *Reachability analysis of systems based on Petri nets*, Košice, TU, 1999, ISBN: 80-88964-07-5

[14] E. Best, R. Devillers, J. Hall, "The petri box calculus: A new causal algebra with multilabel communication", *in Advances in Petri Nets*, Lecture Notes in Computer Science, Vol. 609, pp. 21–69, Springer-Verlag, 1992

Equivalence of table algebras of finite (infinite) tables and corresponding relational calculi

Dmitry Buy
Department of Theory and Technology of Programming
Taras Shevchenko National University of Kyiv
Kyiv, Ukraine
buy@unicyb.kiev.ua

Iryna Glushko
Department of Theory and Technology of Programming
Taras Shevchenko National University of Kyiv
Kyiv, Ukraine
glushkoim@gmail.com

Abstract—**In article two results are presented. The first concerns equivalence of table algebra for infinite tables and corresponding relational calculi, and the second – equivalence of subalgebra of finite tables of the given algebra and corresponding relational calculi in which consideration is limited to only so-called "safe" expressions. The classical relational calculi are filled up by functional and predicate signatures on the universal domain.**

Keywords-relation databases, relation calculus, table algebra

I. INTRODUCTION

Specifying of table in terms of nominal sets is carried out in the monograph [1]. Traditionally the finite set of tuple is understood under the table and the authors take it into account. However, as a rule, mathematical statements about standard properties of specification of relation operations remain true for infinite relations. Further under relation we will understand any set of tuples (with common scheme), in particular infinite.

II. TABLE ALGEBRAS OF FINITE (INFINITE) TABLES

All indefinite concepts and denotations are understood in terms of monograph [1]. Among the two sets that are considered, $\boldsymbol{A}$ is the set of attributes and $\boldsymbol{D}$ is the universal domain.

The table of scheme R ($R \subseteq \boldsymbol{A}$) is pair $\langle t, R\rangle$, where t is set (in particular infinite) of tuples of fixed scheme R. As in the paper [2] a certain scheme is ascribed to every table.

We will designate the set of all tuples (tables) of the scheme R by $S(R)$ ($\boldsymbol{T}(R)$ respectively), and the set of all tuples (tables) by S ($\boldsymbol{T}$ respectively). Hence, $S = \bigcup_{R\subseteq \boldsymbol{A}} S(R)$, $\boldsymbol{T}(R) = P(S(R))$, $\boldsymbol{T} = \bigcup_{R\subseteq A} \boldsymbol{T}(R)$, where $P(A)$ is the power set of a set A.

Under table algebra of infinite tables is understood algebra $\langle \boldsymbol{T}, \Omega_{P,\Xi}\rangle$, where $\boldsymbol{T}$ is the set of all tables, $\Omega_{P,\Xi} = \{\bigcup_R, \bigcap_R, \setminus_R, \sigma_{p,R}, \pi_{X,R}, \underset{R_1,R_2}{\otimes}, \div_{R_2}^{R_1}, Rt_{\xi,R}, \sim_R\}_{X,R,R_1,R_2\subseteq \boldsymbol{A}}^{p\in P,\xi\in\Xi}$ is the signature, P, Ξ are the sets of parameters. The operations of signature $\Omega_{P,\Xi}$ are defined in [2].

In practice we usually work with finite tables. Table algebra considered in the monograph [1] and generalized table algebra represented in the article [2], in which under relation is understood the finite set of tuples, are subalgebras of table algebra of infinite tables. Indeed, if $\langle \boldsymbol{T}', \Omega_{P,\Xi}\rangle$ is generalized table algebra, then set $\boldsymbol{T}'$ is subset of set $\boldsymbol{T}$ and $\boldsymbol{T}'$ is closed under every operation of signature $\Omega_{P,\Xi}$. So, under table algebra of finite tables is understood algebra $\langle \boldsymbol{T}', \Omega_{P,\Xi}\rangle$, where $\boldsymbol{T}'$ is the set of all finite tables. The table of scheme R of table algebra of finite tables is pair $\langle t, R\rangle$, where $t \in T(R)$ is finite table of scheme R. Then $\boldsymbol{T}'(R) = \{\langle t, R\rangle \mid t \in T(R)\}$ is the set of all finite tables on scheme R and $\boldsymbol{T}' = \bigcup_{R\subseteq \boldsymbol{A}} \boldsymbol{T}'(R)$ is the set of all finite tables.

Lemma 1. Take place the following statements:

1) any expression over table algebra for infinite tables can be replaced by equivalent to him expression which uses only operations of selection, join, projection, union, difference and renaming;

2) any expression over table algebra for finite tables can be replaced by equivalent to him expression which uses single-attribute, single-tuple constant tables. □

Proof. For the proof of the first statement we will show that operations of intersection, division, active complement can be expressed through the operations noted in formulation of lemma. Indeed, the following equalities take place:
$\langle t_1, R\rangle \bigcap_R \langle t_2, R\rangle = \langle t_1, R\rangle \setminus_R (\langle t_1, R\rangle \setminus_R \langle t_2, R\rangle)$;
$\langle t_1, R_1\rangle \div_{R_2}^{R_1} \langle t_2, R_2\rangle = \pi_{R',R_1}(\langle t_1, R_1\rangle) \setminus_{R'} \pi_{R',R_1}(\pi_{R',R_1}(\langle t_1, R_1\rangle) \underset{R',R_2}{\otimes} \langle t_2, R_2\rangle \setminus_{R_1} \langle t_1, R_1\rangle)$, where $R_2 \subseteq R_1$, $R' = R_1 \setminus R_2$;
$\sim_R \langle t, R\rangle = C(\langle t, R\rangle) \setminus_R \langle t, R\rangle$,
$C(\langle t, R\rangle) = \pi_{A_1,R}(\langle t, R\rangle) \underset{\{A_1\},\{A_2\}}{\otimes} \dots \underset{\{A_1,\dots,A_{n-1}\},\{A_n\}}{\otimes} \pi_{A_n,R}(\langle t, R\rangle)$, where $R = \{A_1, \dots, A_n\}$ is a scheme of the table $\langle t, R\rangle$ (see, for example, [1], [3]). □

For the proof of the second statement we will replace a constant table $\langle t, R\rangle$ by expression $\bigcup_{i=1}^{n}\left(\{\{\langle A_1, d_{i_1}\rangle\}\} \underset{\{A_1\},\{A_2\}}{\otimes} \ldots \underset{\{A_1,\ldots,A_{m-1}\},\{A_m\}}{\otimes} \{\{\langle A_m, d_{i_m}\rangle\}\}\right)$, where $t = \{s_1,\ldots,s_n\}$, $R = \{A_1,\ldots,A_m\}$ and $s_i = \{\langle A_1, d_{i_1}\rangle,\ldots,\langle A_m, d_{i_m}\rangle\}$, $i = 1,\ldots,n$. □

III. Equivalence of Table Algebra of Infinite Tables and Corresponding Relational Calculi

A. Generalized Tuple Calculus

Relational calculus is the basis of most relational query languages because unlike relational (table) algebra, calculus expresses only what must be the result, and does not determine how to get it. There are two forms of relational calculus: tuple calculus and domain calculus. These forms have been proposed by E. Codd [4] and M. Lacroix with A. Pirotte [5] respectively.

In the classic tuple calculus only binary predicates usually consider and functional signature is generally empty [3, 6]. In the paper [2] tuple calculus is filled up arbitrary predicate and functional signatures on the universal domain $\boldsymbol{D}$. Syntax of terms, atoms and formulas of generalized tuple calculus is defined, the set of legal formulas is introduced; using the concept of free and bound tuple variables, the notions of schema $scheme(\boldsymbol{x}, \boldsymbol{P})$ and a set of attributes $attr(\boldsymbol{x}, \boldsymbol{P})$ with which tuple variable $\boldsymbol{x}$ occurs in a formulas $\boldsymbol{P}$ are put.

As known, the expressions of tuple calculus are built in tuples. The set of all tuples over $\boldsymbol{D}$ is the domain of interpretation of the object variables. Generalized tuple calculus expression has the form $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x})\}$, where

1) formula $\boldsymbol{P}$ is legal;
2) $\boldsymbol{x}$ is the only tuple variable that occurs free in $\boldsymbol{P}$;
3) if $scheme(\boldsymbol{x}, \boldsymbol{P})$ is defined, then $scheme(\boldsymbol{x}, \boldsymbol{P}) = R$, otherwise, $attr(\boldsymbol{x}, \boldsymbol{P}) \subseteq R$.

B. Generalized Domain Calculus

Domain calculus is quite similar to tuple calculus. But there are essential differences. There are no tuple variables in the domain calculus, but there are variables to represent components of tuples, instead. Domain calculus are also supported by the membership condition [7]: $\langle t(\langle A_1, d_1\rangle, \langle A_2, d_2\rangle, \ldots), R\rangle$, where R is scheme, A_i is attribute of table t and d_i is variable of domain or literal (object constants). This condition is true iff in table t there is tuple having specified values over universal domain $\boldsymbol{D}$ for specified attributes.

Domain calculus is also filled up arbitrary predicate and functional signatures on the universal domain $\boldsymbol{D}$ [2]. Syntax of terms, atoms and formulas of generalized domain calculus is defined, the set of legal formulas is introduced. Generalized domain calculus expression has the form $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \mid \boldsymbol{P}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)\}$, where

1. $\boldsymbol{P}$ is a legal domain calculus formula with exactly the free variables $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ (variables over universal domain $\boldsymbol{D}$);
2. $R = \{A_1,\ldots,A_n\}$, $R \subseteq \boldsymbol{A}$ is scheme and the order of attributes is fixed;
3. $scheme(\boldsymbol{x}_i, \boldsymbol{P}) = \boldsymbol{D}$, $i = 1,\ldots,n$.

C. Table Algebra of Infinite Tables, Corresponding Relational Calculi and Theirs Equivalence

Theorem 1. If F is an expression of table algebra of infinite tables, then it is possible effectively to build equivalent to it expression E of generalized tuple calculus [2]. □

Theorem 1 proves that generalized tuple calculus is as expressive as table algebra of infinite tables (in terms of [3]).

Reduction of generalized tuple calculus to generalized domain calculus is conducted [2]. Consider the mapping of the form $E \overset{\varphi}{\mapsto} H$ such that to every expression of generalized tuple calculus puts in correspondence equivalent expression of generalized domain calculus. Consequently, the following theorem takes place.

Theorem 2. If E is an expression of generalized tuple calculus, then it is possible effectively to build equivalent to it expression H of generalized domain calculus. □

So, theorem 2 proves that generalized domain calculus is as expressive as generalized tuple calculus (in terms of [3]).

Thereto, it is proved that table algebra of infinite tables is as expressive as generalized domain calculus.

Theorem 3. If H is an expression of generalized domain calculus, then it is possible effectively to build equivalent to it expression F of table algebra of infinite tables [2]. □

Taking into account the previous theorems we are getting a basic result.

Theorem 4. Table algebra of infinite tables, generalized tuple calculus and generalized domain calculus are equivalent. □

IV. Restricting Generalized Relation Calculus to Yield only Finite Table

Generalized relation calculus allows determining infinite tables. For overcoming of this demerit, inheriting D. Maier [3] and J. D. Ullman [6], we will consider only, so-called, "safe" expressions of relational calculus.

A. Safe Generalized Tuple Calculus Expressions

Let $E = \{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x})\}$ be a generalized tuple calculus expression and $A \in R$. The extended active domain of attribute A relative to $\boldsymbol{P}$ of expression E is the set $D_{A,\boldsymbol{P}} = \{\boldsymbol{d} \mid \boldsymbol{d}$ is a

constant in $\boldsymbol{P}\} \cup \bigcup_{t-table\ from\ \boldsymbol{P}} D_{A,t}$, where $D_{A,t} \overset{def}{=} \{d \mid \exists s(s \in t \,\&\, \langle A,d \rangle \in s)\}$ is the active domain of attribute A relative to table t [1, 3]. Then $D_{\boldsymbol{P}} = \bigcup_{A \in R} D_{A,\boldsymbol{P}}$.

Generalized tuple calculus expression $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x})\}$ is safe if the following three conditions hold.

1) If formula $\boldsymbol{P}(s/\boldsymbol{x})$ is true, then $s(A) \in D_{\boldsymbol{P}}$, $A \in R$;
2) For every subformula of $\boldsymbol{P}$ of the form $\exists \boldsymbol{y}(X)\boldsymbol{G}(\boldsymbol{y},z_1,z_2,\ldots,z_k)$, $\boldsymbol{G}(s/\boldsymbol{y},u_1/z_1,u_2/z_2,\ldots,u_k/z_k)$ is true implies $s(A) \in D_{\boldsymbol{G}}$, $A \in X$ and $X \subseteq R$, where $\boldsymbol{y},z_1,z_2,\ldots,z_k$ are all the free tuple variables in $\boldsymbol{G}$;
3) For every subformula of $\boldsymbol{P}$ of the form $\forall \boldsymbol{y}(X)\boldsymbol{G}(\boldsymbol{y},z_1,z_2,\ldots,z_k)$, $s(A) \notin D_{\boldsymbol{G}}$, $A \in X$ implies $\boldsymbol{G}(s/\boldsymbol{y},u_1/z_1,u_2/z_2,\ldots,u_k/z_k)$ is true, where $\boldsymbol{y},z_1,z_2,\ldots,z_k$ are all the free tuple variables in $\boldsymbol{G}$.

The conditions 2) and 3) allow determining the truth of a quantified formulas $\exists \boldsymbol{y}(X)\boldsymbol{G}(\boldsymbol{y},z_1,z_2,\ldots,z_k)$ and $\forall \boldsymbol{y}(X)\boldsymbol{G}(\boldsymbol{y},z_1,z_2,\ldots,z_k)$ by considering only those values of variable $\boldsymbol{y}$ that are the functions of the form $X \to D_{\boldsymbol{G}}$.

B. Safe Generalization Domain Calculus Expressions

As in the case of generalized tuple calculus we will introduce the concept of extended active domain of attribute A. Let $H = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \mid \boldsymbol{P}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)\}$ be a generalized domain calculus expression, $A \in \boldsymbol{A}$. The extended active domain of attribute A relative to $\boldsymbol{P}$ of expression E is the set $D_{A,\boldsymbol{P}} = \{\boldsymbol{d} \mid \boldsymbol{d}$ is a constant in $\boldsymbol{P}\} \cup \bigcup_{t-table\ from\ \boldsymbol{P}} D_{A,t}$, where $D_{A,t}$ is active domain of attribute A.

Generalized domain calculus expression $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \mid \boldsymbol{P}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)\}$ is safe if the following three conditions hold.

1) If for constants $d_1,\ldots,d_n$, formula $\boldsymbol{P}(d_1/\boldsymbol{x}_1,\ldots,d_n/\boldsymbol{x}_n)$ is true, then $d_i \in D_{A_i,\boldsymbol{P}}$, $i = 1,\ldots,n$, where A_i is attribute from R which is associated with variable $\boldsymbol{x}_i$, $R = \{A_1,\ldots,A_n\}$;
2) If $\exists \boldsymbol{y}(A)$ $\boldsymbol{G}$ is a subformula of $\boldsymbol{P}$, then formula $\boldsymbol{G}(d/\boldsymbol{y})$ is true implies that $d \in D_{A,\boldsymbol{G}}$, $X \subseteq R$;
3) If $\forall \boldsymbol{y}(A)$ $\boldsymbol{G}$ is a subformula of $\boldsymbol{P}$, then $d \notin D_{A,\boldsymbol{G}}$ implies that formula $\boldsymbol{G}(d/\boldsymbol{y})$ is true.

At the limited interpretation of the expression $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \mid \boldsymbol{P}(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)\}$, the values of variable $\boldsymbol{x}_i$ are values from $D_{A_i,\boldsymbol{P}}$, $i = 1,\ldots,n$.

C. Table Algebra of Finite Tables,Corresponding Relational Calculi and theirs equivalence

Theorem 5. If F is an expression of table algebra of finite tables, then it is possible effectively to build equivalent to it safe generalized tuple calculus expression E. □

Proof. According to the lemma 1 for proof of the theorem we consider expressions of table algebra of finite tables which use only single-attribute, single-tuple constant relations and the operations of union, difference, selection, projection, join and rename only.

The proof proceeds by induction on the number of operators in F.

Basis (no operators). There are two cases. Firstly, $F = \langle t,R \rangle$, where $t \in T(R)$, then $E = \{\boldsymbol{x}(R) \mid t(\boldsymbol{x})\}$. E is a safe expression, since any tuple $\boldsymbol{x}$ satisfying $t(\boldsymbol{x})$ is in $\langle t,R \rangle$, whereupon $\boldsymbol{x}(A) \in D_t$ for $A \in R$. Secondly, F is the constant table $t = \{\{\langle A,d \rangle\}\}$, then $E = \{\boldsymbol{x}(\{A\}) \mid \boldsymbol{x}(A) = d\}$.

Induction. Assume the theorem holds for any table algebra expression with fewer than k operators. Let F has k operators.

Case 1. $F = F_1 \cup_R F_2$. F_1 and F_2 each have less than k operators, and by the inductive hypothesis we can find generalized tuple calculus expressions $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x})\}$ and $\{\boldsymbol{x}(R) \mid \boldsymbol{Q}(\boldsymbol{x})\}$ equivalent to F_1 and F_2 respectively. Then E is equivalent to $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x}) \vee \boldsymbol{Q}(\boldsymbol{x})\}$. We will show that this expression is safe. If $\boldsymbol{x}$ satisfies $\boldsymbol{P}(\boldsymbol{x}) \vee \boldsymbol{Q}(\boldsymbol{x})$, then $\boldsymbol{x}(A) \in D_{\boldsymbol{P}}$ or $\boldsymbol{x}(A) \in D_{\boldsymbol{Q}}$. It is not hardly to check, that $D_{\boldsymbol{P} \vee \boldsymbol{Q}} = D_{\boldsymbol{P}} \cup D_{\boldsymbol{Q}}$. Hence, formula $\boldsymbol{P}(\boldsymbol{x}) \vee \boldsymbol{Q}(\boldsymbol{x})$ is true only when $\boldsymbol{x}(A) \in D_{\boldsymbol{P} \vee \boldsymbol{Q}}$. Thereto, any subformula $\exists \boldsymbol{y}(X)\boldsymbol{G}(\boldsymbol{y},z_1,z_2,\ldots,z_k)$ (or $\forall \boldsymbol{y}(X)\boldsymbol{G}(\boldsymbol{y},z_1,z_2,\ldots,z_k)$) in $\boldsymbol{P}(\boldsymbol{x}) \vee \boldsymbol{Q}(\boldsymbol{x})$ must be within $\boldsymbol{P}(\boldsymbol{x})$ or $\boldsymbol{Q}(\boldsymbol{x})$. So the inductive hypothesis assures that these subformulas do not violate safety.□

Case 2. $F = F_1 \setminus_R F_2$. Then generalized tuple calculus expressions $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x})\}$ and $\{\boldsymbol{x}(R) \mid \boldsymbol{Q}(\boldsymbol{x})\}$ equivalent to F_1 and F_2 respectively exist as in case 1. Then E is $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x}) \wedge \neg \boldsymbol{Q}(\boldsymbol{x})\}$. Proof of safety is brought like a previous case with use of equality $D_{\boldsymbol{P} \wedge \neg \boldsymbol{Q}} = D_{\boldsymbol{P}} \cup D_{\boldsymbol{Q}}$. □

Case 3. $F = \sigma_{\tilde{p},R}(F_1)$. Let $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x})\}$ be safe generalized tuple calculus expression equivalent to F_1. Then E is $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x}) \wedge \boldsymbol{p}(\boldsymbol{x}(A_1),\ldots,\boldsymbol{x}(A_m))\}$, where $R = \{A_1,\ldots,A_m\}$ is scheme of table, that is the value of expression F_1. Assume that predicate-parameter of selection is defined as $\tilde{p}(s) = true \Leftrightarrow \boldsymbol{p}(s(A_1),\ldots,s(A_m)) = true$, $s \in S(R)$, where $\boldsymbol{p}$ is signature predicate symbol of arity m. The expression E is safe, because each tuple $\boldsymbol{x}$ is restricted to those values to which formula $\boldsymbol{P}$ restricts the tuple. □

Case 4. $F = \pi_{X,R}(F_1)$. Let $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x})\}$ be safe generalized tuple calculus expression equivalent to F_1. Then E is $\{\boldsymbol{y}(X \cap R) \mid \exists \boldsymbol{x}(R)(\boldsymbol{P}(\boldsymbol{x}) \wedge \underset{A \in X \cap R}{\wedge} \boldsymbol{y}(A) = \boldsymbol{x}(A))\}$. This expression is safe, because $\boldsymbol{y}(A)$ is restricted to values that $\boldsymbol{x}(A)$ may take, $A \in X \cap R$. □

Case 5. $F = F_1 \underset{R_1,R_2}{\otimes} F_2$. Let $\{\boldsymbol{x}(R_1) \mid \boldsymbol{P}(\boldsymbol{x})\}$ and $\{\boldsymbol{y}(R_2) \mid \boldsymbol{Q}(\boldsymbol{y})\}$ be safe generalized tuple calculus expressions equivalent to F_1 and F_2 respectively. Then E is $\{\boldsymbol{z}(R_1 \cup R_2) \mid \exists \boldsymbol{x}(R_1) \exists \boldsymbol{y}(R_2)(\boldsymbol{P}(\boldsymbol{x}) \wedge \boldsymbol{Q}(\boldsymbol{y}) \wedge \underset{A \in R_1}{\wedge} \boldsymbol{z}(A) = \boldsymbol{x}(A) \wedge \underset{A \in R_2}{\wedge} \boldsymbol{z}(A) = \boldsymbol{y}(A))\}$. This expression is safe, because $\boldsymbol{z}(A)$ is restricted to values that $\boldsymbol{x}(A)$, $A \in R_1$, and $\boldsymbol{y}(A)$, $A \in R_2$, may take. □

Case 6. $F = Rt_{\xi,R}(F_1)$, where $\xi : \boldsymbol{A} \xrightarrow{\sim} \boldsymbol{A}$ is injective function that renames attributes. Safe generalized tuple calculus expression $\{\boldsymbol{x}(R) \mid \boldsymbol{P}(\boldsymbol{x})\}$ equivalent to F_1 exists. Then E is $\{\boldsymbol{y}(R_2) \mid \exists \boldsymbol{x}(R)(\boldsymbol{P}(\boldsymbol{x}) \wedge \underset{C \in R \setminus dom\xi}{\wedge} \boldsymbol{y}(C) = \boldsymbol{x}(C) \wedge \underset{A \in R \cap dom\xi}{\wedge} \boldsymbol{x}(A) = \boldsymbol{y}(\xi(A)))\}$, where $R_2 = (R \setminus dom\xi) \cup \xi[R]$ (in essence, this is a scheme of the renamed table). This expression is safe, because $\boldsymbol{y}(C)$ is restricted to values that $\boldsymbol{x}(C)$ may take, $C \in R \setminus dom\xi$, and $\boldsymbol{y}(\xi(A))$ is restricted to values that $\boldsymbol{x}(A)$ may take, $A \in R \cap dom\xi$. □□

Consider the mapping of the form $E \overset{\varphi}{\mapsto} H$ such that every expression of generalized tuple calculus puts in correspondence equivalent expression of generalized domain calculus. Let $E = \{\boldsymbol{y}(R) \mid \boldsymbol{P}(\boldsymbol{y})\}$ be safe generalized tuple calculus expression, tuple $\boldsymbol{y}$ is the only tuple variable that occurs free in $\boldsymbol{P}$ and $R = \{A_1,...,A_n\}$. Make the necessary replacements (see [2]) and obtain the generalized domain calculus expression $H = \{\boldsymbol{y}_1,...,\boldsymbol{y}_n \mid \boldsymbol{P}(\boldsymbol{y}_1,...,\boldsymbol{y}_n)\}$. It should also be clear that the values that may be assumed by every variables $\boldsymbol{z}_i$ of generalized domain calculus are exactly those that could be assumed by $\boldsymbol{z}(\boldsymbol{A}_i)$ in the original expression. Thus if E is safe, so is the resulting domain calculus expression and value of expression H coincides with the value of expression E. That is why takes place the following theorem.

Theorem 6. If E is a safe generalized tuple calculus expression then it is possible effectively to build equivalent to it safe generalized domain calculus expression H. □

Theorem 7. If H is a safe generalized domain calculus expression then it is possible effectively to build equivalent to it expression F of table algebra of finite tables. □

Proof. Let $H = \{\boldsymbol{x}_1,...,\boldsymbol{x}_n \mid \boldsymbol{P}(\boldsymbol{x}_1,...,\boldsymbol{x}_n)\}$ be a safe generalized domain calculus expression, where $R = \{A_1,...,A_n\}$ is scheme of the table, that is the value of expression H. Although H is safe expression, there can be subformulas of formulas $\boldsymbol{P}(\boldsymbol{x}_1,...,\boldsymbol{x}_n)$ which do not have this property. We will prove by induction on the number of operators in a subformula $\boldsymbol{G}$ of $\boldsymbol{P}$ that if $\boldsymbol{G}$ has free domain variables $\boldsymbol{y}_1,...,\boldsymbol{y}_m$, then $\underset{A_i \in R_G}{N} D_{A_i,\boldsymbol{P}} \cap_{R_G} \{\boldsymbol{y}_1,...,\boldsymbol{y}_m \mid \boldsymbol{G}(\boldsymbol{y}_1,...,\boldsymbol{y}_m)\}$ has an equivalent expression in table algebra of finite tables F_G. Then, as a special case, when $\boldsymbol{G}$ is $\boldsymbol{P}$ itself, we have an algebraic expression for $\underset{A_j \in R}{N} D_{A_j,\boldsymbol{P}} \cap_R \{\boldsymbol{x}_1,...,\boldsymbol{x}_n \mid \boldsymbol{P}(\boldsymbol{x}_1,...,\boldsymbol{x}_n)\}$. It is assumed that $\boldsymbol{y}_1,...,\boldsymbol{y}_m$ are the only tuple variables that occurs free in $\boldsymbol{G}$ and the table of scheme $R_G = \{A_1,...,A_m\}$ is a value of expression $\underset{A_i \in R_G}{N} D_{A_i,\boldsymbol{P}} \cap_{R_G} \{\boldsymbol{y}_1,...,\boldsymbol{y}_m \mid \boldsymbol{G}(\boldsymbol{y}_1,...,\boldsymbol{y}_m)\}$.

Replace domain variables in $\boldsymbol{P}$ so that no variable is bound in two places or occurs both free and bound in $\boldsymbol{P}$. Note that every variable is associated with an attribute, either by a quantifier $\forall \boldsymbol{x}(\boldsymbol{A})$ or $\exists \boldsymbol{x}(\boldsymbol{A})$ if a variable is bound in $\boldsymbol{P}$ or by appearing to the left of the bar in the expression H, if a variable is free in $\boldsymbol{P}$.

For any attribute A we will map algebraic expression. A single-attribute table $\langle t, \{A\} \rangle$ is a value of this algebraic expression. This table contains all tuples of kind $s = \{\langle A, d_i \rangle\}$, $d_i \in D_{A,\boldsymbol{P}}$. We will designate this algebraic expression through $[D]$. Consider all possible cases.

Basis (no operators). Subformula $\boldsymbol{G}$ is an atom of the form $\boldsymbol{p}(\boldsymbol{u}_1,...,\boldsymbol{u}_m)$ or $t(\boldsymbol{a}_1,...,\boldsymbol{a}_n)$.

1) Let $\boldsymbol{G}$ be $\boldsymbol{p}(\boldsymbol{u}_1,...,\boldsymbol{u}_m)$, where $\boldsymbol{u}_i$ are the terms of generalized domain calculus and $\boldsymbol{y}_1,...,\boldsymbol{y}_k$ are the all variables of this terms. Then F_G is $\sigma_{\tilde{p}}([D]_1 \underset{R_1,R_2}{\otimes} ... \underset{R_1 \cup .. \cup R_{k-1},R_k}{\otimes} [D]_k)$, where R_i are the single-attribute schemes of tables and those tables are the values of algebraic expressions $[D]_i$, $i = 1,...,k$. Assume that predicate-parameter of select is defined as $\tilde{p}(s) = true \Leftrightarrow \boldsymbol{p}(s(A_1),...,s(A_m)) = true, \quad s \in S(R)$, where $\boldsymbol{p}$ is signature predicate symbol of arity m. Expressions $[D]_i$ are built on the attribute associated with a variable $\boldsymbol{y}_i$, $i = 1,...,k$. □

2) Let $\boldsymbol{G}$ be $t(\boldsymbol{a}_1,...,\boldsymbol{a}_k)$, where $\boldsymbol{a}_i$ is either a constant or a variable over the universal domain $\boldsymbol{D}$. Let $R = \{C_1,...,C_k\}$ be a scheme of table $\langle t, R \rangle$. The algebraic expression F_G is $\pi_X(\sigma_{\tilde{p}}(\langle t, R \rangle))$, where $\tilde{p}$ is predicate-parameter of select which is a conjunction of comparisons $C_i = \boldsymbol{a}_i$ for each $\boldsymbol{a}_i$ that is a constant; X is the set of attributes $\{C_j \mid \boldsymbol{a}_j$ is a variable$\}$. If the set of attributes X is empty, then there is no any projection in expression F_G. □

Induction. Assume $\boldsymbol{G}$ has at least one operator and that the inductive hypothesis is true for all subformulas of $\boldsymbol{P}$ having fewer operators than $\boldsymbol{G}$.

Case 1. $\boldsymbol{G} = \neg\boldsymbol{Q}$. Let F_Q be the algebraic expression for $\boldsymbol{Q}$ of kind $\underset{A_i \in R_Q}{N} D_{A_i,\boldsymbol{P}} \cap_{R_Q} \{\boldsymbol{y}_1,...,\boldsymbol{y}_m \mid \boldsymbol{Q}(\boldsymbol{y}_1,...,\boldsymbol{y}_m)\}$ and the table of scheme R_Q is a value of expression F_Q. Then $F_G = \sim_{R_Q} F_Q$. This algebraic expression is equivalent to $\underset{A_i \in R_Q}{N} D_{A_i,\boldsymbol{P}} \setminus_{R_Q} \{\boldsymbol{y}_1,...,\boldsymbol{y}_m \mid \boldsymbol{Q}(\boldsymbol{y}_1,...,\boldsymbol{y}_m)\}$, which also is equivalent to expression $\underset{A_i \in R_G}{N} D_{A_i,\boldsymbol{P}} \cap_{R_G} \{\boldsymbol{y}_1,...,\boldsymbol{y}_m \mid \neg\boldsymbol{Q}(\boldsymbol{y}_1,...,\boldsymbol{y}_m)\}$. □

Case 2. $\boldsymbol{G} = \boldsymbol{Q} \vee \boldsymbol{Q}'$. Let $\boldsymbol{Q}$ has free variables $z_1,...,z_k, v_1,...,v_p$ and let $\boldsymbol{Q}'$ has free variables $z_1,...,z_k, w_1,...,w_q$, where $v_1,...,v_p$ and $w_1,...,w_q$ are distinct. F_Q and $F_{Q'}$ are the algebraic expressions for $\boldsymbol{Q}$ and $\boldsymbol{Q}'$ respectively of kind $\underset{A_i \in R_Q}{N} D_{A_i,\boldsymbol{P}} \cap_{R_Q} \{\boldsymbol{z}_1,...,\boldsymbol{z}_k, \boldsymbol{v}_1,\ldots,\boldsymbol{v}_p \mid \boldsymbol{Q}(z_1,...,z_k, v_1,\ldots,v_p)\}$ and $\underset{A_s \in R_{Q'}}{N} D_{A_s,\boldsymbol{P}} \cap_{R_{Q'}} \{\boldsymbol{z}_1,...,\boldsymbol{z}_k, \boldsymbol{w}_1,\ldots,\boldsymbol{w}_q \mid \boldsymbol{Q}'(z_1,...,z_k, w_1,\ldots,w_q)\}$, where R_Q and $R_{Q'}$ are the schemes of the tables that are the values of those algebraic expressions. Let C_i, $i = 1,...,p$ be the attributes associated with variables $v_1,...,v_p$ and let K_j, $j = 1,...,q$ be the attributes associated with variables $w_1,...,w_q$. Let $F_1 = F_Q \underset{R_Q,\{K_1\}}{\otimes} [D]_1 \underset{R_Q \cup \{K_1\},\{K_2\}}{\otimes} ... \underset{R_Q \cup \{K_1,...,K_{q-1}\},\{K_q\}}{\otimes} [D]_q$ and let $F_2 = F_{Q'} \underset{R_{Q'},\{C_1\}}{\otimes} [D]'_1 \underset{R_{Q'} \cup \{C_1\},\{C_2\}}{\otimes} ... \underset{R_{Q'} \cup \{C_1,...,C_{p-1}\},\{C_p\}}{\otimes} [D]'_p$. Expression $[D]'_i$, $i = 1,...,p$ is built on the attribute C_i and expression $[D]_j$, $j = 1,...,q$ is built on the attribute K_j.

Let R_{F_1} and R_{F_2} be the schemes of the tables that are the values of algebraic expressions F_1 and F_2 respectively. Note that $R_{F_1} = R_{F_2}$. Then $F_G = F_1 \bigcup_{R_{F_1}} F_2$. Expression F_1 is equal to $\underset{A_i \in R_G}{N} D_{A_i,\boldsymbol{P}} \cap_{R_G} \{\boldsymbol{y}_1,...,\boldsymbol{y}_m \mid \boldsymbol{Q}(\boldsymbol{z}_1,...,\boldsymbol{z}_k, \boldsymbol{v}_1,\ldots,\boldsymbol{v}_p)\}$ and expression F_2 is equal to $\underset{A_i \in R_G}{N} D_{A_i,\boldsymbol{P}} \cap_{R_G} \{\boldsymbol{y}_1,...,\boldsymbol{y}_m \mid \boldsymbol{Q}'(\boldsymbol{z}_1,...,\boldsymbol{z}_k, \boldsymbol{w}_1,\ldots,\boldsymbol{w}_q)\}$, that's why expression F_G is equal to $\underset{A_i \in R_G}{N} D_{A_i,\boldsymbol{P}} \cap_{R_G} \{\boldsymbol{y}_1,...,\boldsymbol{y}_m \mid \boldsymbol{G}(\boldsymbol{y}_1,...,\boldsymbol{y}_m)\}$, $i = 1,\ldots,m$. □

Case 3. $\boldsymbol{G} = \boldsymbol{Q} \wedge \boldsymbol{Q}'$. This generalized domain calculus expression can be replaced by $\boldsymbol{G} = \neg(\neg\boldsymbol{Q} \vee \neg\boldsymbol{Q}')$ (De Morgan's law).□

Case 4. $\boldsymbol{G} = \exists\boldsymbol{y}_{m+1}(A)\boldsymbol{Q}$. Let F_Q be the algebraic expression for $\boldsymbol{Q}$ of kind $\underset{A_i \in R_Q}{N} D_{A_i,\boldsymbol{P}} \cap_{R_Q} \{\boldsymbol{y}_1,...,\boldsymbol{y}_{m+1} \mid \boldsymbol{Q}(\boldsymbol{y}_1,...,\boldsymbol{y}_{m+1})\}$, where R_Q is the scheme of the table that is the value of algebraic expression F_Q. Since $\boldsymbol{P}$ is the formula of the safe generalized domain calculus expression, and therefore $\boldsymbol{Q}(\boldsymbol{y}_1,...,\boldsymbol{y}_{m+1})$ is never true unless $\boldsymbol{y}_{m+1}(A)$ is in the set $D_{A,\boldsymbol{Q}}$, which is a subset of $D_{A,\boldsymbol{P}}$. Then F_G is $\pi_{X \setminus \{A\},X}(F_Q)$, where X is the scheme of the table that is the value of algebraic expression F_Q. This expression is equal to $\underset{A_i \in R_G}{N} D_{A_i,\boldsymbol{P}} \cap_{R_G} \{\boldsymbol{y}_1,...,\boldsymbol{y}_m \mid \exists\boldsymbol{y}_{m+1}(A)\boldsymbol{Q}(\boldsymbol{y}_1,...,\boldsymbol{y}_{m+1})\}$. □

Case 5. $\boldsymbol{G} = \forall\boldsymbol{x}(A)\boldsymbol{Q}$. This generalized domain calculus expression can be replaced by $\forall\boldsymbol{x}(A)\boldsymbol{Q} = \neg(\exists\boldsymbol{x}(A))(\neg\boldsymbol{Q})$. □□

Taking into account the theorems 5, 6, 7 we are getting a basic result.

Theorem 8. Table algebra of finite tables, limited generalized tuple calculus and limited generalized domain calculus are equivalent. □

V. Conclusions

In article table algebras for infinite and finite tables are considered. The classical relational calculi are filled up by functional and predicate signatures on the universal domain (while usually consider only binary predicates and functional signature is generally empty). On the one hand, it is proved the equivalence of table algebra for infinite tables and corresponding relational calculi, and on the other hand, it is showed the equivalence of subalgebra of finite tables of the given algebra and corresponding relational calculi in which consideration is limited to only so-called "safe" expressions.

References

[1] V. Redko, J. Brona, D. Buy, S. Poliakov, "Relation Database: Relation Algebras and SQL-similar Languages,". Kyiv: 2001. (in Ukrainian)

[2] D. Buy, I. Glushko, "Generalized Table Algebra, Generalized Tuple Calculus, Generalized Domain Calculus and theirs Equivalence", Bulletin of Kyiv National Taras Shevchenko University, series:Physical and mathematical sciences, Vol. 1, Kyiv, 2011, pp. 86-95. (in Ukrainian)

[3] D. Maier, The theory of relational databases, Rockvill, Maryland, 1983.

[4] E. F. Codd, "Relational Completeness of Data Base Sublanguages," in Data Base Systems. Prentice-Hall, New York, 1972, pp. 65-93.

[5] M. Lacroix, A. Pirotte, "Domain-oriented relational languages,". Proc. 3rd Int. Conf. on Very Large Data Bases. Tokyo, 1977, pp. 370-378.

[6] J. D. Ullman, Principles of Database Systems, Rockvill, Maryland, 1982.

[7] C. J. Date, An Introduction to Database Systems, 8th ed., Adisson Wesley, 2005.

Intrusion Detection System Epistème

Daniel Mihályi, Valerie Novitzká, Martina Ľaľová
Technical University, Department of Computer Science and Informatics,
Letná 9, Košice, Slovakia
Daniel.Mihalyi@tuke.sk, Valerie.Novitzka@tuke.sk, Martina.Lalova@tuke.sk

***Abstract*—In our paper we investigate the possibilities of modal logics in coalgebras of program systems. We deal with simplified model of intrusion detection system. We model intrusion detection system as a coalgebra and construct its Kripke model of coalgebraic modal linear logic using powerset endofunctor. In this model we present our idea how a fragment of epistemic linear logic can provide knowledge and belief of intrusion attempt.**
***Index Terms*– coalgebra, epistemic logic, linear logic, Kripke model**

I. Introduction

In programming systems we are interested not only about their construction, but also in their observable behavior. Observable behavior can be modeled by coalgebras [1], [9], [10] using modal logic [3]. Coalgebras can model various types of transition systems. Within behavioral observation some events can repeat and they can provide us some interesting knowledge about program systems. Following the results in [11] we can be sure that objective knowledge implies rational belief. Knowledge and belief are fundamental notions of epistemic logic.

In our approach we investigate possibilities of obtaining objective knowledge and rational belief for simplified model of intrusion detection system (IDS). Incoming packets form infinite streams and some of them can contain some intrusion attempts. These attemps can be recognized through characteristical symptoms. Determined combination of these symptoms give us a knowledge about some kind of incoming intrusion. Moreover, if it incomes from the same IP address and repeatedly, then we are sure that it is a real intrusion attempt and we can make our decision about competent reactions. We use a fragment of epistemic linear logic with explicit logical operators for objective knowledge and rational belief as a suitable logical system for reasoning about intrusion attempts.

In this paper we go out from our results published in [4] where IDS is modeled as a coalgebra over appropriate polynomial endofunctor. We show how knowledge and belief can be formulated in Kripke model of possible worlds over this coalgebra.

II. Coalgebraic modal linear logic for IDS

Typically [3], coalgebraic approach uses a modal logic with two modal operators ($\Box$ for necessity and $\Diamond$ for possibility). In our approach we work with modal linear logic fragment because of the causality of its linear implication. We use the syntax of this fragment:

$$\varphi ::= \ a_i \mid \varphi_1 \multimap \varphi_2 \mid \varphi_1 \otimes \varphi_2 \mid \Box\varphi \mid \Diamond\varphi \mid \mathbf{1} \qquad (1)$$

where

- a_i are atomic propositions,
- $\varphi_1 \multimap \varphi_2$ means linear implication. This implication ensures that the action φ_2 follows after the action φ_1,
- $\varphi_1 \otimes \varphi_2$ is multiplicative conjuction expressing that φ_1 and φ_2 are both executed,
- $\Box\varphi$ means application of the necessity operator to formula φ,
- $\Diamond\varphi$ means application of the possibility operator to formula φ,
- $\mathbf{1}$ is a neutral element of the multiplicative conjuction.

We illustrate coalgebraic modal logic on the example of IDS. We consider only two types of possible intrusions, A or B. Let O be a sender indentification (e.g. its IP address). Then we construct the category $\mathcal{P}acket$ of incoming packets as follows:

- objects are significant packet fragments for identification of intrusion attempts,

$$p = (A + B) \times O \qquad (2)$$

- morphisms are mappings $next$ between objects

$$next : p_i \to p_{i+1} \qquad (3)$$

where $i \in \mathbb{N}$. It is clear that the category $\mathcal{P}acket$ has special sets as objects. Now we define polynomial endofunctor $T : \mathcal{P}acket \to \mathcal{P}acket$ on this category as follows:

$$T(p) = X \times p \quad and \quad T(next(p)) = X \times next(p). \qquad (4)$$

Then coalgebraic specification for polynomial endofunctor T is

$$\langle hd, tl\rangle : \rho_p \to T\rho_p. \qquad (5)$$

In [4] we modeled IDS system as a coalgebra

$$(\rho_p, \langle hd, tl\rangle) \qquad (6)$$

for infinite packet stream ρ_p. The operations hd resp. tl are obvious operations returning head resp. tail of a given stream.

Contemporary experiences in the area of system behaviour have shown the importance of selection an appropriate modal logical language as a specification language for various transition systems. Formulae of this language are used to logical

reasoning over states of dynamic system that are captured by the coalgebra of corresponding polynomial (powerset) endofunctor. We formulated coalgebraic logic based on multimodal language that can be suitable for behavioral description of infinite, non trivial heterogenous data structures, i.e. packets at the coalgebra as intrusion detection system in [5].

In the following text let $Prop$ be the set of propositions.

As a model of our logic we use Kripke model of possible worlds [12] that is characterized by Kripke frame

$$(W, \leq, w_0) \tag{7}$$

with satisfaction relation $\models$ as a tuple

$$(W, \leq, \models, w_0) \tag{8}$$

where

- W is a set of possible worlds,
- $\leq$ is accessibility relation $\leq \quad \subseteq \quad W \times W$,
- $\models$ is satisfaction relation

$$\models: W \times Prop \to \{0, 1\} \tag{9}$$

  where 1 means satisfaction and 0 means non satisfaction,
- w_0 is a designated world.

Notation $w_1 \leq w_2$ we read as follows: a possible world w_2 is reachable (accessible) from w_1. According to the philosophy of possible world semantics: "it is possible what is reachable together" [12].

A coalgebra can be seen as a general form of Kripke semantics for modal logic. An interpretation of a formula in coalgebra is given by predicate lifting. Predicate lifting is a natural transformation

$$\lambda : \mathcal{P}^- \Rightarrow \mathcal{P}^- \circ T \tag{10}$$

where $\mathcal{P}^-$ is a contravariant powerset functor $\mathcal{P}^- : Set \to Set$ between sets (Fig. 1).

$$\begin{array}{c} (\mathcal{P}^- \circ T)(\rho_p) \\ \Uparrow \\ \lambda(\rho_p) \\ \Uparrow \\ \mathcal{P}^-(\rho_p) \end{array}$$

Fig. 1. Predicate lifting

$\lambda(\rho_p)$ is a class of morphisms defined by

$$\lambda(\rho_p) : \mathcal{P}^-(\rho_p) \to (\mathcal{P}^- \circ T^{sp})(\rho_p). \tag{11}$$

Now we can interpret the formulae in any T-model as follows:

$$(\lambda(\rho_p), \langle hd, tl\rangle : \rho_p \to T\rho_p) \tag{12}$$

where

- for every formula φ we define validity set $[\![\varphi]\!] \subset \rho_p$ by induction on the structure of φ,
- for modal operator $\Box$ we define the validity as

$$[\![\Box\varphi]\!] = \mathcal{P}^-(\langle hd, tl\rangle : \rho_p \to T\rho_p) \circ \lambda([\![\varphi]\!]). \tag{13}$$

The operator of possibility $\diamondsuit$ is dual to the operator of necessity $\Box$. In the following we use the traditional notation for Kripke models. It is clear that the set W of possible worlds corresponds with the stream of packets ρ_p.

- Every world $w \in W$ corresponds with a packet $p \in \rho_p$,
- the reachability relation

$$\leq \quad \subseteq \quad W \times W \tag{14}$$

  gives a $\mathcal{P}$-coalgebra

$$(W, \langle hd, tl\rangle : \rho_p \to T\rho_p\rangle) \tag{15}$$

  where

$$(\langle hd, tl\rangle : \rho_p \to T\rho_p)_{\leq}(w) = \{w' \in W \mid (w, w') \in \leq\}. \tag{16}$$

Then the formulae of coalgebraic modal language are expressed as the infinite sequence

$$\begin{array}{r} (\mathbf{1}) \\ (p_1, \mathbf{1}) \\ (p_1, (p_2, \mathbf{1})) \\ (p_1, (p_2, (p_3, \mathbf{1}))) \\ \dots \\ \bigotimes \{(p_0, p_1, p_2, p_3, \dots, (true))\} \end{array} \tag{17}$$

where the raw $(\mathbf{1})$ denotes the empty formula and $\bigotimes$ denotes infinite linear multiplicative conjuction.

III. Epistemic linear logic for IDS

Traditionally, epistemic logic is characterised as intensional logic with modalities treating with objective knowledge and rational belief [11]. We work with the following fragment of epistemic linear logic [2]:

$$\varphi ::= \ a_i \mid K_c\varphi \mid B_c\varphi \mid \varphi_1 \otimes \varphi_2 \mid \varphi_1 \multimap \varphi_2 \mid !\varphi \tag{18}$$

where

- a_i are atomic propositions (i.e. pieces of knowledge),
- $K_c\varphi$ denotes that a rational agent c knows that φ,
- $B_c\varphi$ denotes that a rational agent c believes that φ,
- $\varphi_1 \otimes \varphi_2$ realizes the linear conjunction of two formulas φ_1,φ_2,
- $\varphi_1 \multimap \varphi_2$ realizes the linear implication of two formulas φ_1,φ_2,
- $!\varphi$ is empiric modal operator expressing repeated objective knowledge.

IV. Motivation Example

According example mentioned in [4], an infinite stream of packets ρ_p can be realised by the following infinite sequence

TABLE I
PARTICULAR TYPES OF NETWORK INTRUSIONS

Type A	Type B
(ICMP Ping NMAP)	*(TCP Portscan)*
`IP Protocol == icmp`	`MAC Addr == MACDAD`
`dsize == 0`	`IP Protocol == 255`
`itype == 8`	`IP TTL == 0`

TABLE II
INTRUSION TYPE A - $ICMP\ Ping\ NMAP$

Type A	a_1	a_2	a_3
w_8	0	0	0
w_7	0	0	1
w_6	0	1	0
w_5	0	1	1
w_4	1	0	0
w_3	1	0	1
w_2	1	1	0
w_1	1	1	1

$$
\begin{aligned}
&(p_1, p_2, p_3, p_4, p_5 \ldots) \mapsto \\
&\mapsto (A \times O, (B \times O, \mathbf{1} \times O, A \times O, B \times O, \ldots)) \quad \mapsto \\
&\mapsto (A \times O, B \times (O, \mathbf{1} \times O, A \times O, B \times O, \ldots)) \quad \mapsto \\
&\mapsto (A \times O, B \times O, \mathbf{1} \times O, (A \times O, B \times O, \ldots)) \quad \mapsto \\
&\mapsto (A \times O, B \times O, \mathbf{1} \times O, A \times O, (B \times O, \ldots)) \quad \mapsto \\
&\mapsto \ldots
\end{aligned} \tag{19}
$$

where p_1, p_2, p_3, p_4, p_5 are treated packet fragments from any patern of network traffic and A resp. B are specifications of particular intrusion attempt *ICMP Ping NMAP* resp. *TCP Portscan* mentioned in TABLE I.

For our fragment of epistemic linear logic we define its model as Kripke model of possible worlds. We can use the same Kripke frame

$$(W, \leq, w_0) \tag{20}$$

constructed for modal linear logic in the previous section.

Let $AP = \{a_1, a_2, a_3, b_1, b_2, b_3, \ldots\}$ be the set of atomic propositions. Every atomic proposition denotes one symptom of appropriate intrusion attempt. According to the Table I we denote the knowledge about intrusion attempt of "Type A" i.e. $ICMP\ Ping\ NMAP$ by the tuple (a_1, a_2, a_3) where

- $a_1 : IPProtocol$ is equal to $icmp$,
- $a_2 : dsize$ is equal to 0,
- $a_3 : itype$ is equal to 8.

If a symptom a_i is present then we assign the value 1 to it. Otherwise we assign to a_i the value 0. According to the TABLE II we will work with eight possible worlds.
The intrusion attempt of "Type A" occurs only if all a_i have the value 1. Therefore we consider w_1 as designated world ($w_1 \equiv w_0$).

Similarly, for the next intrusion attempt of "Type B", i.e. $TCPPortscan$ we consider the following pieces of knowledge (b_1, b_2, b_3)

- $b_1 : MACAddr$ is equal to $MACDAD$,
- $b_2 : IPProtocol$ is equal to 255,

TABLE III
INTRUSION TYPE B - $TCP\ Portscan$

Type N	b_1	b_2	b_3
w_{16}	0	0	0
w_{15}	0	0	1
w_{14}	0	1	0
w_{13}	0	1	1
w_{12}	1	0	0
w_{11}	1	0	1
w_{10}	1	1	0
w_9	1	1	1

- $b_3 : IPTTL$ is equal to 0.

If a symptom b_i is present then we assign the value 1 to it. Otherwise we assign to b_i the value 0. According to the TABLE III we will work also with eight possible worlds.
The intrusion attempt of "Type B" occurs only if all b_i have the value 1. Therefore we consider w_9 as designated world ($w_9 \equiv w_0$).

In our simple example we consider only one (rational) agent 007. This agent can be a part of communication interface between human and computer system.
From the behavioral sequence of packets (19) our agent can achieve particular knowledge (for example based on visible alerts on monitor). We denote by

- $K_{007}a_i$ that our agent 007 has the particular piece of knowledge about a_i,
- $K_{007}b_i$ that our agent 007 has the particular piece of knowledge about b_i.

By induction

- $K_{007}\varphi$ is an epistemic formula that denotes

$$K_{007}a_1 \otimes K_{007}a_2 \otimes K_{007}a_3 \tag{21}$$

- $K_{007}\psi$ is an epistemic formula that denotes

$$K_{007}b_1 \otimes K_{007}b_2 \otimes K_{007}b_3 \tag{22}$$

In Fig. 2. we illustrate the process of achieving knowledge about intrusion attempts of "Type A" and "Type B". Then

- $K_{007}\varphi$ denotes objective knowledge that on the packet p_1 was captured intrusion attempt of "Type A" i.e. *ICMP Ping NMAP*,
- $K_{007}\psi$ denotes objective knowledge that on the packet p_3 was captured intrusion attempt of "Type B" i.e. *TCP Portscan*.

Here we need to acquire another piece of knowledge about sender of a given packet e.g. its IP Address. This knowledge can be achieve from the attribute *srcIP* in packet header. We denote by $K_{007}\tau$ the objective knowledge about sender identification of the "catched packet".

The formula

$$(K_{007}\varphi \multimap K_{007}\psi) \otimes K_{007}\tau \tag{23}$$

describes the situation that an intrusion attempt of "Type A" followed by an intrusion attempt of "Type B" have occured from an intruder with the same identification. This kind of attempt is known as vertical portscan.

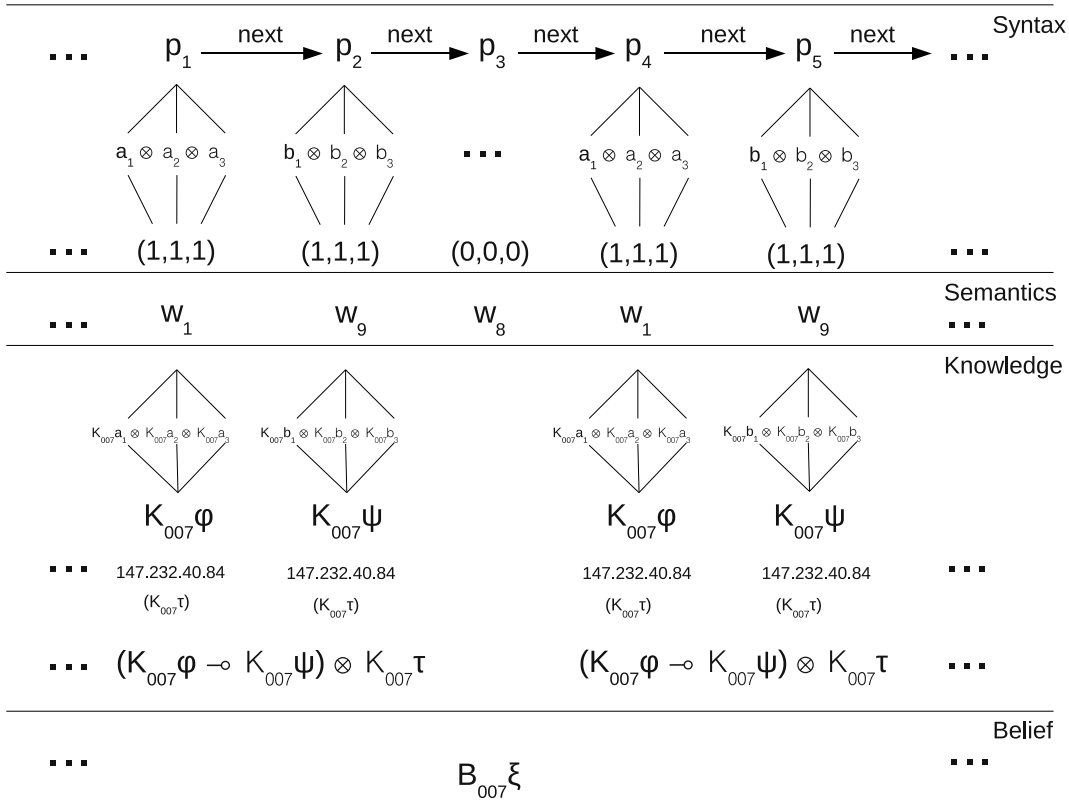


Fig. 2. Episteme

We achieve empirical rational belief denoted $B_{007}\xi$ from repeatedly occuring knowledge about intrusion attempts of "Type A" and "Type B" incomming from the same sender that is given by its identification O.

$$\begin{array}{ll} \texttt{if} & K_{007}\varphi \otimes K_{007}\psi \otimes K_{007}\tau \quad \texttt{and} \\ & K_{007}\varphi \otimes K_{007}\psi \otimes K_{007}\tau \quad \texttt{and} \\ & \ldots \\ \texttt{then} & B_{007}\xi \end{array} \tag{24}$$

Using empiric modality operator we can denote the process from objective knowledge to rational belief as the following linear implication

$$!(K_{007}\varphi \otimes K_{007}\psi \otimes K_{007}\tau) \multimap B_{007}\xi \tag{25}$$

V. Conclusion

In this paper we present our ideas about achieving knowledge and belief from the observable behaviour of program systems. We illustrate our approach on the simplified intrusion detection system and show how the pices of knowledge can be achieved from some symptoms, how its combination gives us the knowledge about some intrusion detection and how the repeating of some knowledge leads to belief about some concrete intrusion attempt. Our approach is based on coalgebraic modeling of system behavior. Instead of obvious correspondence with modal logic we construct a model of simple fragment of epistemic linear logic suitable for our purposes and we show the process from pieces of knowledge to knowledge and belief on this model.

Our approach use only IP protocol version ipv4 and only two possible intrusion attempts. Our idea will be generalized for any type of intrusion attempt and we would like to investigate achieving of knowledge and belief for IP protocol version ipv6, too. In further research we would like to investigate distributed intrusion attempts using groups of agents.

VI. Acknowledgements

This work is the result of the project implementation: Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

References

[1] Gumm H. P.: Functors for Coalgebras. Algebra Universalis, Vol. 45, 135–147 (2001)
[2] Kamide, N.: Linear and affine logics with temporal, spatial and epistemic operators. Theoretical Computer Science 353 (13), pp. 165207. (2006)
[3] KURZ, A.: Coalgebras and Modal Logic. CWI, Amsterdam, Netherlands (2001)

[4] Mihályi D., Novitzká V.: A Coalgebra as an Intrusion Detection System. Acta Polytechnica Hungarica, Budapest, Vol. 7, Issue 2, ISSN 1785-8860, 71–79 (2010)
[5] Mihályi D., Novitzká V.: Princípy duality medzi konštruovaním a správaním programov. Equilibria, ISBN 9788089284580 (2010)
[6] Moss, L. S.: Coalgebraic Logic. Annals of Pure and Applied Logic, Vol. 96 (1999)
[7] Schubert Ch.: Topo-Bisimulations are Coalgebraic. Rendiconti dell' Istituto di Matemàtica dell'Universitá di Trieste, Vol. 42, ISSN 0049-4704, 257–270 (2010)
[8] Schröder L., Pattinson D.: Coalgebraic Modal Logic: Forays Beyond Rank 1. IFIP WG 1.3 meeting, Sierra Nevada, (2008)
[9] Slodičák, V., and Macko, P.: New approaches in functional programming using algebras and coalgebras. In European Joint Conferrences on Theory and Practice of Software - ETAPS 2011 (March 2011),Workshop on Generative Technologies, Universität des Saarlandes, Saarbrücken, Germany, pp. pp. 1323. ISBN 978-963-284-188-5.
[10] Slodičák V.: Some useful structures for categorical approach for program behavior, Journal of Information and Organizational Sciences, Vol. 35, No. 1, 2011, pp. 99-109, ISSN 1846-9418
[11] Voorbraak F.: Generalized Kripke Models For Epistemic Logic. Proceedings of Fourth Conference on Theoretical aspects of reasoning about knowledge, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, SBN:1-55860-243-9, 214–228 (1992)
[12] Zouhar M.: Základy logiky. Proceedings of Fourth Conference on Theoretical aspects of reasoning about knowledge, Veda SAV, Bratislava ISBN 978-80-224-1040-3 (2008)

Multisets: Operations, Partial Order, Computability, Application

Dmitry Buy
Department of Theory and Technology of Programming
Taras Shevchenko National University of Kyiv
Kyiv, Ukraine
buy@unicyb.kiev.ua

Juliya Bogatyreva
Department of Theory and Technology of Programming
Taras Shevchenko National University of Kyiv
Kyiv, Ukraine
jbogatyreva@gmail.com

Abstract—**This article is devoted to selected questions of multisets theory. The basic definitions of multisets theory are given. Structure of multisets family is considered. Computability on finite sets and multisets of natural numbers is defined.**

Keywords-multiset; lattice; complete lattice; computibility; primitive programming algebra

I. INTRODUCTION

Multiset is a family of arbitrary elements, which can be duplicated.

Despite presence of numerous bibliography of multiset theory and its application in various fields (see, for example, papers [1], [2] and authors papers [3], [7], [8], [10], [11], [14], [18], [19], [20]) it is too early for saying about presence of complete multiset theory. The paper is devoted to developing of multisets theory: (1) about properties of multiset operations and relations between these operations; (2) about structure of multisets family ordered by natural inclusion relation; (3) about computability (numeric according to Malcev) on finite multisets.

II. BASIC DIFINITION

Formally, multiset α with a basis U_α is a function $\alpha : U_\alpha \to N^+$, where U_α is an arbitrary set (in a classical Cantor's understanding), and $N^+ \overset{def}{=} \{1,2,...\}$ is a set of natural numbers without zero.

Characteristic function of multiset α (see, for example, [22]) is a function $\chi_\alpha : D \to N$, values of which are specified by the following piecewise schema:

$$\chi_\alpha(d) = \begin{cases} \alpha(d), if\ d \in dom\ \alpha, \\ 0, else; \end{cases}$$

for all $d \in D$, where D is the universe of elements of multiset bases.

Let's introduce a binary inclusion relation over multisets:

$$\beta \preceq \alpha \overset{def}{\Leftrightarrow} U_\beta \subseteq U_\alpha \ \&\ \forall d\left(d \in U_\beta \Rightarrow \beta(d) \le \alpha(d)\right).$$

Directly from definition follows that this relation is a partial order.

Analogs of standard set-theoretic operations and operations, which are using peculiarities of multisets and, therefore cannot be useful for (abstract) sets, are defined.

Properties of introduced operations are considered: characteristic of inclusion relation in terms of intersection and union operations; idempotency, commutativity, associativity, mutual distributivity of the operations, absorption laws, analogs of law of double negation and De Morgan law [12].

Let's define only operations of multiset union and multiset intersection. Operation $\bigcup_{All}$ (respectively $\bigcap_{All}$) compares multisets α and β with multiset, characteristic function value of which on arbitrary argument d is specified by the following expression: $\max(\chi_\alpha(d), \chi_\beta(d))$ (respectively $\min(\chi_\alpha(d), \chi_\beta(d))$).

III. STRUCTURE OF MULTISETS FAMILY

Let's construct a multiset lattice and nest it into two complete lattices.

Lemma 1. Operations $\bigcup_{All}$ and $\bigcap_{All}$ are idempotent, commutative, associative.□

Lemma 2. Following absorption laws are true for arbitrary multisets: $\alpha \bigcap_{All} (\alpha \bigcup_{All} \beta) = \alpha$, $\alpha \bigcup_{All} (\alpha \bigcap_{All} \beta) = \alpha$.□

Set of all multisets (of some universe of elements of multiset bases) designates as M.

Theorem 1. Partially ordered set $\langle \mathrm{M}, \preceq \rangle$ is a lattice, and $\sup_{\preceq}\{\alpha,\beta\} = \alpha \bigcup_{All} \beta$, $\inf_{\preceq}\{\alpha,\beta\} = \alpha \bigcap_{All} \beta$.□

Let's adduce more considerable results about structure of multisets family [1], [3], [4], [7], [8], [15], [16], [17], [19], [21].

Statement 1 (about structure of partially ordered set of multisets). The following statements are valid:

1) empty multiset $\varnothing_m$ (characteristic function of this multiset is a constant function, value of which is everywhere equal zero) is the least element in $\langle M, \preceq \rangle$;

2) $\inf \mu = \alpha$; for arbitrary not empty set of multisets $\mu \subseteq \mathrm{M}$; here characteristic function of multiset α is specified by expression $\chi_\alpha(d) = \min_{\beta \in \mu} \chi_\beta(d)$;
3) for arbitrary set of multisets μ :
   supremum μ exists $\Leftrightarrow$ μ is upper-bounded;
4) $\sup \mu = \alpha$, where μ is arbitrary set of multisets, which have the least upper bound, and characteristic function of multiset α is specified by expression $\chi_\alpha(d) = \max_{\beta \in \mu} \chi_\beta(d)$.□

Theorem 2. Partially ordered set $\langle M, \preceq \rangle$ is a conditionally complete set and a complete semilattice, and the least upper bound (infimum) and the greatest lower bound (supremum) are specified according to formulas of the statement 1.□

Let's add the greatest element T to partially ordered set $\langle M, \preceq \rangle$; then derived partially ordered set is designated as $\langle M \cup \{T\}, \preceq \rangle$.

Corollary 1. Partially ordered set $\langle M \cup \{T\}, \preceq \rangle$ is a complete lattice with the least element $\varnothing_m$ and the greatest element T .□

The last result is an informative one, that is why let's consider nesting into another complete lattice, which has a considerable (formal) interpretation for defining a semantics of recursive queries in SQL-similar language [5], [9]. For this reason let's generalize the definition of multiset by allowing arbitrary multiplicity of elements of basis.

Let's extend the notion of multisets. To that end, let's add set of natural numbers without zero with the greatest element ∞ and assume $N_\infty^+ = N^+ \cup \{\infty\}$. Then multiset will be a function $\alpha : U_\alpha \to N_\infty^+$; a family of all those multisets is defined as M_∞ . Order on set N_∞^+ is designated as $\leq_\infty$, then order on multisets $\preceq$ is extended to $\alpha \preceq_\infty \beta \overset{def}{\Leftrightarrow} \overset{def}{\Leftrightarrow} \forall d \left(\chi_\alpha(d) \leq_\infty \chi_\beta(d)\right)$, $\alpha, \beta \in M_\infty$. Let's mention, that orders $\preceq$, $\leq$ are constructed via direct product of partially ordered sets $\langle N, \leq \rangle$ and $\langle N_\infty, \leq_\infty \rangle$ correspondently.

Theorem 3. Partially ordered set $\langle M_\infty, \preceq_\infty \rangle$ is a complete lattice with the least element $\varnothing_m$ and the greatest element T_∞ , where $T_\infty : D \to \{\infty\}$, $T_\infty(d) = \infty$ for all d . The greatest lower bounds are defined by formulas of the statement 1, and for the least upper bounds formula $\sup \mu = \alpha$ is valid, where characteristic function of multiset α is following: $\chi_\alpha(d) = \sup_{\leq_\infty, \beta \in \mu} \chi_\beta(d)$.□

IV. Computibility on Multisets

Let's consider (numeric) computability on finite sets and multisets of natural numbers. Primitive program algebras (PPA) which has been introduced by V.N. Redko in the article [23] are using as an instrument for the definition of a computable functions class.

In papers [6], [13] systems of the generators of sets PPA and multisets PPA were constructed.

System of the generators of set PPA Σ consists of equality predicate $X = Y$, set union function $X \cup Y$, functions of addition $X \oplus Y$ and subtraction $X \div Y$ (finite) sets (of natural numbers), selector functions I_m^n and constant functions $\{1\}(X)$ and $\varnothing(X)$, which are correspondently fixing singleton $\{1\}$ and empty set on arbitrary argument:

$$\Sigma \overset{def}{=} \left\{X = Y,\ X \cup Y,\ X \oplus Y,\ X \div Y,\ \{1\}(X),\ \varnothing(X),\ I_m^n\right\}_{m=1,\ldots,n}^{n=1,2,\ldots}.$$

Theorem 4. System Σ is a system of the generators of sets PPA.□

System of the generators of multisets PPA Σ_{M} consists of multiset equality predicate $\alpha = \beta$, functions of union $\alpha \cup_{All} \beta$, addition $\alpha \oplus \beta$ and subtraction $\alpha \div \beta$ of multisets (of natural numbers), constant functions $\{1^1\}(\alpha)$ and $\varnothing_m(\alpha)$, which are correspondently fixing multiset $\{1^1\} \overset{def}{=} \{\langle 1, 1 \rangle\}$ and empty multiset, special binary function on multisets $\varphi(\alpha, \beta)$ and selector functions I_m^n :

$$\Sigma_{\mathrm{M}} \overset{def}{=} \{\alpha = \beta,\ \alpha \cup_{All} \beta,\ \alpha \oplus \beta,\ \alpha \div \beta,$$
$$\{1^1\}(\alpha),\ \varnothing_m(\alpha),\ \varphi(\alpha, \beta),\ I_m^n\}_{m=1,\ldots,n}^{n=1,2,\ldots},$$

where for function φ should only the following equality hold true $\varphi(\{n_1^1\}, \{k_1^1\}) = \{n_1^{k_1}\}$ (and for $k_1 = 0$ $\varphi(\{n_1^1\}, \{k_1^1\}) = \varnothing_m$).

Theorem 5. System Σ_{M} is a system of the generators of multisets PPA.□

Conclusions

The considered results indicate about intentionality and depth of multisets theory, which is undoubtedly incomplete.

Therewith, multisets theory can be developing via alternative way.

Indeed, considered multisets theory is based on multiset definition as a function from some Cantor's set (basis) into set of natural numbers without zero (every element from basis is compare to its finite multiplicity). However, it is possible to define multiset as a partition or as an equivalence relation (interpretation: "same" elements of multiset are in the same equivalence class).

Let's note that in this case multisets with arbitrary multiplicity of elements immediately appear.

Developing of multisets theory constructed according to the last definition requires separate research and will be considered in next articles.

References

[1] W.D. Blizard, "The development of multiset theory", Notre Dame Journal of Formal Logic, Vol. 30, No. 1, 1989, pp. 36-66.

[2] G. Lamperti, M. Melchiori, M. Zanella, "On multisets in database systems", Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View, number 2235 in Lecture Notes in Computing Science, Berlin: Springer-Verlag, 2001, pp. 147-215.

[3] D. Buy, J. Bogatyreva, "Methods of multiset lattice construction", Papers of 9th International Conference on Applied Mathematics, February 2–5, 2010, Bratislava, Slovakia, pp. 407-413.

[4] D. Buy, J. Bogatyreva, "Structure of partially ordered family of multisets", Proceedings of CSE 2010 International Scientific Conference on Computer Science and Engineering, September 20–22, 2010, Košice – Stará Ľubovňa, Slovakia, pp. 40-43.

[5] D. Buy, S. Polyakov, "Recursive queries in SQL and their generalization – system of recursive queries", Proceedings of CSE 2010 International Scientific Conference on Computer Science and Engineering, September 20–22, 2010, Košice – Stará Ľubovňa, Slovakia, pp. 252-257.

[6] J. Bogatyreva, "Computability on finite sets and multisets", Bulletin of Kyiv National Taras Shevchenko University, series : Physical and mathematical sciences, Vol. 4, Kyiv, Ukraine, 2010, pp. 88-96. (in Ukrainian)

[7] J. Bogatyreva, "Multisets: bibliography, multisets lattice", Proceedings of 6-th International Scientific Conference "Theoretical and Applied Aspects of Program System Development" – TAAPSD'2009, December 8-10, 2009, Kyiv, Ukraine, Vol. 2, pp. 13-20. (in Russian)

[8] J. Bogatyreva, "Multisets: bibliographical review, multiset lattice construction", Journal "Problems of programming", Vol. 2-3, Kyiv, Special edition, 2010, pp. 68-71. (in Russian)

[9] D. Buy, S. Polyakov, "Composition semantic of recursive queries in SQL-similar language", Bulletin of Kyiv National Taras Shevchenko University, series : Physical and mathematical sciences, Vol. 1, Kyiv, 2010, pp. 45-50. (in Ukrainian)

[10] D. Buy, J. Bogatyreva, "Multisets is an alternative of set-theoretical platform in mathematic substantiation of information technology", Journal "Information Model of Knowledge", Vol. 19, ITHEA, 2010, pp. 377-386. (in Russian)

[11] D. Buy, J. Bogatyreva, "Multisets", Proceedings of the XVI International Scientific Conference "Problems of Decision Making under Uncertainties" – PDMU'2010, October 4–8, 2010, Yalta, Ukraine, pp. 149. (in Russian)

[12] D. Buy, J. Bogatyreva, "Multisets: definition, operations, basic properties", Proceedings of 5-st International Scientific Conference "Theoretical and Applied Aspects of Program System Development" – TAAPSD'2008, September 22–26, 2008, Kyiv, Ukraine, pp. 23-26. (in Ukrainian)

[13] D. Buy, J. Bogatyreva, "Computability on multisets", Proceedings of International Scientific and Practical Conference devoted to 80-anniversary of physico-mathematical department of State Pedagogical University of Kirovograd, November 26, 2010, pp. 28-30. (in Ukrainian)

[14] D. Buy, J. Bogatyreva, "Bibliographical review of multisets theory", Journal "Scientific notes NAUKMA", series : Computer science, Vol. 112, Kyiv, Ukraine, 2010, pp. 4-9. (in Ukrainian)

[15] D. Buy, J. Bogatyreva, "Multisets lattice", Proceedings of International Scientific Conference "Integrate models and soft computing in artificial intelligence", May 28-30, 2009, Kolomna, Russia, electronic resources: raai.org/resurs/papers/kolomna2009/doklad/ Bui_Bogatyreva.doc. (in Russian)

[16] D. Buy, J. Bogatyreva, "Structure of multisets family", Proceedings of 7-st International Scientific Conference "Theoretical and Applied Aspects of Program System Development" – TAAPSD'2010, October 4-8, 2010, Kyiv, Ukraine, pp. 121-125. (in Ukrainian)

[17] D. Buy, J. Bogatyreva, "Structure of partially ordered multisets family", Journal "Information Model of Knowledge", Vol. 19, ITHEA, 2010, pp. 387-391. (in Russian)

[18] D. Buy, J. Bogatyreva, "Modern state of multisets theory", Bulletin of Kyiv National Taras Shevchenko University, series : Physical and mathematical sciences, Vol. 1, Kyiv, Ukraine, 2010, pp. 51-58. (in Ukrainian)

[19] D. Buy, J. Bogatyreva, "Multisets theory: bibliography, application in databases", Journal "Radioelectronics and computer systems", Vol. 7(48), 2010, pp. 56-62. (in Russian)

[20] D. Buy, J. Bogatyreva, "Formal model of DNA-computation" Proceedings of the 1-st Congress "Medical and Biological informatics and cybernetic", June 23-26, 2010, Kyiv, Ukraine, pp. 221. (in Ukrainian)

[21] D. Buy, J. Bogatyreva, "To problem of multisets lattice", Proceedings of X International Seminar "Discrete mathematics and its application", February 1-6, 2010, Moskva, Russian, pp. 220-222. (in Russian)

[22] V. Redko, J. Brona, D. Buy, S. Poliakov, "Relation database: relation algebras and sql-similar languages", Kyiv, 2001. (in Ukrainian)

[23] V. Redko, "Universe programming logics and its application" Proceedings of all-USSR symposium about theoretical and systemic programming, Shtiica, 1983, p. 310-326. (in Russian)

On applying action semantics

Viliam Slodičák, Valerie Novitzká, Pavol Macko
Department of Computers and Informatics
Technical University of Košice, Faculty of Electrical Engineering and Informatics
Letná 9, 042 00 Košice, Slovak Republic
Email: Viliam.Slodicak@tuke.sk, Valerie.Novitzka@tuke.sk, Pavol.Macko@tuke.sk

***Abstract*—In this paper we deal with the application of action semantics. We present a short introduction into foundations of action semantics and its application in functional paradigm. The foundations for the action semantics descriptions of functional programs are formulated in this paper. Reached results we demonstrate on the concrete well-known example from informatics - the Fibonacci numbers. The computation of Fibonacci numbers have been implemented in the object-oriented functional language OCaml and the descrition of the program is expressed in action semantics.**

***Index Terms*—Action semantics, functional paradigm, actions, semantical description.**

I. Introduction

The modern computability theory has its roots in the work done at the beginning of the twentieth century to formalise the concept of an "algorithm" without referring to a specific programming language or physical computational device [2]. Knowing and proving of the expected behavior of complex program systems is very important and actual *rôle* [6], [15], [18]. Programmers understand programming languages in terms of basic concepts such as control flow, bindings, modifications of storage, and parameter passing. Formal specifications often obscure these notions to the point that the reader must invest considerable time to determine whether a language follows static or dynamic scoping and how parameters are actually passed. Sometimes the most fundamental concepts of the programming language are the hardest to understand in a formal definition [19].

Action semantics, which attempts to answer these criticisms of formal methods for language specification, has been developed over the past few years. It is a framework for the formal description of programming languages. Action semantics is fully equivalent with other semantic methods, like denotational semantics, operational semantics or axiomatic semantics. Its main advantage over other frameworks is pragmatic: action-semantic descriptions (ASDs) can scale up easy to real programming languages [1], [7], [12], [20]. This is due to the inherent extensibility and modifiability of ASDs, ensuring that extensions and changes to the described language require only proportionate changes in its description. On the other hand, adding an unforeseen construct to a language may require a reformulation of the entire description in denotational or operational semantics expressed in [8], [9].

In the chapter II we describe fundamentals of action semantics. In the next chapter we formulate some basic principles of action semantics in functional paradigm. Reached results we demonstrate on the example of calculating the Fibonacci numbers which is implemented in the OCaml language [4] and then described in action semantics. We apply the results from the previous works [16], [17] about the recursion. These descriptions are very simple and understandable. Because action semantics is expressed as natural language, the main advantage is that anyone who is familiar with described language is able to understand action semantics descriptions very easily.

II. Action semantics

The framework of action semantics has been initially developed at the University of Aarhus by Peter D. Mosses, in collaboration with David Watt from University of Glasgow. Action semantics deals with three kinds of semantic entities: actions, yielders and data. Fundamentals of action semantics are actions which are essentially dynamic computational entities. They incorporate the performance of computational behavior, using values passed to them to generate new values that reflect changes in the state of the computation. So the performance of an action directly represents the information of processing the behavior and reflects the gradual, step-wise nature of computation: each step of an action performance may access and/or change the current information. Other semantic entities used in action semantics are yielders and data. The actions are main kind of entities, the yielders and data are subsidiary. The notation used for specifying actions and the subsidiary semantic entities is called action notation [7]. In action semantics, the semantics of a programming language is defined by mapping program phrases to actions. The performance of these actions relates closely to the execution of the program phrases. Primitive actions can store data in storage cells, bind identifiers to data, compute values, test truth values, and so on [13].

A performance of an action which may be part of an enclosing action either:

- *completes*, corresponding to normal termination;
- *escapes*, corresponding to exceptional termination;
- *fails*, corresponding to abandoning an alternative;
- *diverges*, corresponding to deadlock.

The different kinds of information give rise to so called *facets* of actions which have been classified according to [7]. They are focusing on the processing of at most one kind of information at a time:

- *the basic facet*, processing independently of information (control flows);

- *the functional facet*, processing transient information (actions are given and give data);
- *the declarative facet*, processing scoped information (actions receive and produce bindings);
- *the imperative facet*, processing stable information (actions reserve and dispose cells of storage, and change the data stored in cells);
- *the communicative facet*, processing permanent information (actions send messages, receive messages in buffers, and offer contracts to agents) [7].

The standard notation for specifying actions consists of primitive actions and action combinators. Examples of action combinators are depicted on the following figures: action combinator `and` (Fig. 1), action combinator `and then` (Fig. 2), action combinator `then` (Fig. 3) and action combinator `or` (Fig. 4). In diagrams, the scoped information flows from left to the right whereas transients flow from top to the bottom.

- A_1 `and` A_2 (Fig. 1) allows the performance of the two actions to be interleaved. No control dependency in diagram, so actions can be performed collaterally.
- A_1 `and then` A_2 (Fig. 2) perform the first action and then perform the second.
- A_1 `then` A_2 (Fig. 3) perform the first action using the transients given to the combined action and then perform the second action using the transients given by the first action. The transients given by the combined action are the transients given by the second action.
- A_1 `or` A_2 (Fig. 4) arbitrarily choose one of the subactions and perform it with given transients and received bindings. If the chosen action fails, perform the other subaction with original transients and bindings.

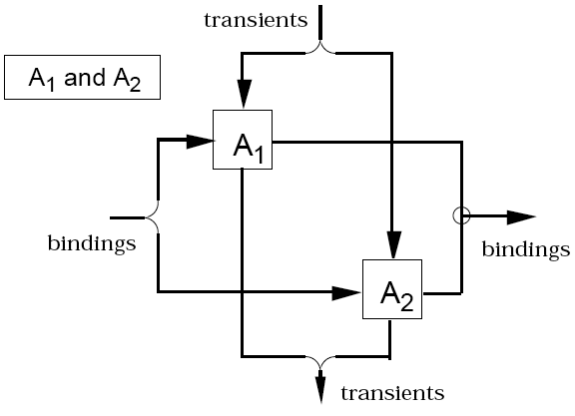


Fig. 1. Action combinator A_1 *and* A_2

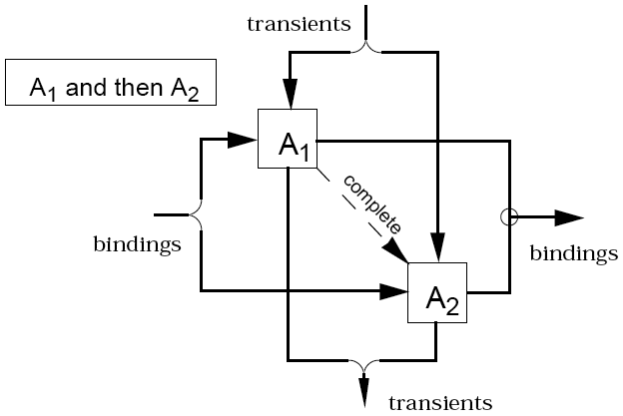


Fig. 2. Action combinator A_1 *and then* A_2

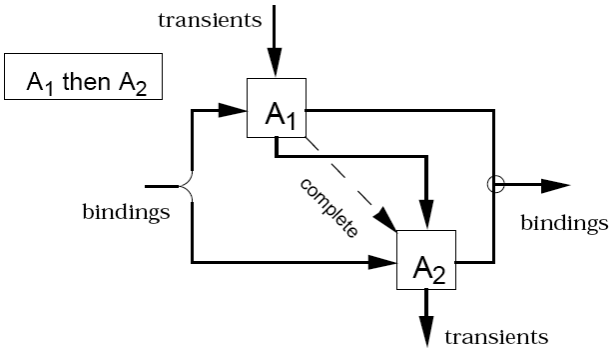


Fig. 3. Action combinator A_1 *then* A_2

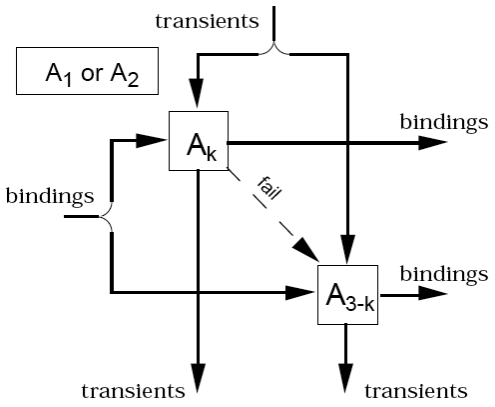


Fig. 4. Action combinator A_1 *or* A_2

The data entities consist of mathematical values, such as integers, Boolean values, and abstract cells representing memory locations, that embody particles of information. Sorts of data used by action semantics are defined by algebraic specifications. Yielders encompass unevaluated pieces of data whose values depend on the current information incorporating the state of the computation. Yielders are occurring in actions and may access, but they are not allowed to change the current information.

The standard notation for specifying actions consists of primitive actions and action combinators. Action combinators combine existing actions, normally using infix notation, to

control the order which subactions are performed in as well as the data flow to and from their subactions. Action combinators are used to define sequential, selective, iterative, and block structuring control flow as well as to manage the flow of information between actions. The standard symbols used in action notation are ordinary English words. In fact, action notation is very near to natural language:

- terms standing for actions form imperative verb phrases involving conjunctions and adverbs, e.g. `check it and then escape`;
- terms standing for data and yielders form noun phrases, e.g. `the items of the given list`.

These simple principles for choice of symbols provide a surprisingly grammatical fragment of English, allowing specifications of actions to be made fluently readable. The informal appearance and suggestive words of action notation should encourage programmers to read it. Compared to other formalisms, such as λ-notation, action notation may appear to lack conciseness: each symbol generally consists of several letters, rather than a single sign. But the comparison should also take into account that each action combinator usually corresponds to a complex pattern of applications and abstractions in λ-notation. In any case, the increased length of each symbol seems to be far outweighed by its increased perspicuity.

III. Action semantics in functional paradigm

Action semantics can be successfully used also for the description of functional programs [10], [11]. In action semantics we use generally three main actions for the description of programming languages:

- *execute* - used for executing of statements;
- *elaborate* - used with declarations;
- *evaluate* - used for evaluating expressions.

In functional paradigm we use only two main actions: *evaluate* and *elaborate*. Action *execute* is not important in functional paradigm. Typical for functional programs is that they do not deal with storage. Therefore we will not use actions of imperative facet for allocating memory locations, storing values and getting values from cells in memory in our action semantics descriptions of functional programs.

Important for functional paradigm is an evaluating of the expressions and elaborating functions. To allow referring them in the program code, they are associated to names (identifiers). These associations are called bindings. A binding can be *global*, when declared at the top level of the source code, or *local*, when declared in a *let* or *letrec* expressions that contain it. The difference between *let* and *letrec* expressions is that in the latter mutual recursion is allowed. We provide this description of evaluation of simple expression:

elaborate⟦`let` I`:Var = E:Expression`⟧ =
  evaluate ⟦ `E` ⟧
    `then bind` I `to the given value`

After declaration we are able to use it anytime in our program. The value is bound to its identifier, so we can get the value of this expression simply by using *evaluate* action:

evaluate⟦ I`:Var` ⟧ =
  `give the value bound to` I

Description of function with one argument should seem like this:

elaborate⟦`let` I_f`:Var` I_{p1}`:Var = E:Expression`⟧ =
  evaluate⟦`E`⟧
    `then bind` I_f `to the given value`

In the expression E the parameter of the function is used. The value of the function we can get simply by action *evaluate*:

evaluate⟦ I_{p1}`:Var` ⟧ =
  `give the value bound to` I_{p1}

General definition for function with different number of arguments:

elaborate⟦`let` I_f`:Var <` I_p`:Var >`$^+$ `= E:Expression`⟧ =
  evaluate⟦`E`⟧
    `then`
  `bind` I_f `to the given value`

IV. Example

We present the description of functional program in action semantics at the well-known algorithm: Fibonacci numbers. In mathematics, the Fibonacci numbers are the numbers in the integer sequence where the first two Fibonacci numbers are 0 and 1 (sometimes first two Fibonacci numbers are considered 1 and 1), and each subsequent number is the sum of the previous two [3], [5], [14].

The construction of sequence:

- F_0 = 0, sequence is (0);
- F_1 = 1, sequence is (0, 1);
- $F_2 = F_0 + F_1 = 0 + 1 = 1$, sequence is (0, 1, 1);
- $F_3 = F_1 + F_2 = 1 + 1 = 2$, sequence is (0, 1, 1, 2);
- $F_4 = F_2 + F_3 = 1 + 2 = 3$, sequence is (0, 1, 1, 2, 3);
- $F_5 = F_3 + F_4 = 2 + 3 = 5$, sequence is (0, 1, 1, 2, 3, 5);
- *etc.* ...

In mathematical terms we define the Fibonacci sequence by the linear recurrence relation, for $n > 1$:

$$F_n = F_{n-2} + F_{n-1}$$

The function for the computation of Fibonacci numbers in the OCaml language has the form:

```
let rec fibDnC(n) =
  if (n==0 || n==1) then 1
  else fibDnC(n-1) + fibDnC(n-2);;
```

This function is constructed by the "Divide et Impera" method.

A. Description in action semantics

We use a substitution for the term that calculates the Fibonacci number.

Let E = `if` ($n == 0$ || $n == 1$) `then 1`
  `else` $fibDnC(n$ - 1) + $fibDnC(n$ - 2)

Next we elaborate the function $fibDnC(n)$

elaborate ⟦ `let rec` *fibDnC (n)* = E⟧ =
`recursively bind` *fibDnC* `to`
`closure of`
`abstraction of`
evaluate⟦E⟧ =

`recursively bind` *fibDnC* `to`
`closure of`
`abstraction of`
evaluate⟦n⟧
`and then`
`give the` $TruthValue$ `of`
`(the given number is equal`
`to the number` 0
`or`
`the given number is equal`
`to the number` 1)
`then`
`check the given` $TruthValue$
`and then`
`give the number` 1
`or`
`check not the given` $TruthValue$
`and then`
`give the sum of`
(**evaluate**⟦$fibDnC(n-1)$⟧
`and`
evaluate⟦$fibDnC(n-2)$⟧)

The value of argument n in function $fibDnC(n)$ we get in action **evaluate**:

evaluate ⟦n⟧ =
give the value bound to n

B. The description of an example in Action semantics

We show an elaboration of function $fibDnC(n)$ for the input argument $n = 5$.

evaluate ⟦*fibDnC (n)* = E⟧$s\,[n \mapsto 5]$ =
`give the value bound to`
`closure of`
`abstraction of`
evaluate⟦E⟧$s\,[n \mapsto 5]$ =

`give the value bound to`
`closure of`
`abstraction of`
`give the number 5`
`and then`
`give the` $TruthValue$ `of`
`(the given number is equal`
`to the number` 0
`or`
`the given number is equal`
`to the number` 1)
`then`
`check not the given` $TruthValue$
`and then`
`give the sum of`
(**evaluate**⟦$fibDnC(n)$⟧$s\,[n \mapsto 4]$
`and`
evaluate⟦$fibDnC(n)$⟧$s\,[n \mapsto 3]$)

Next we evaluate the function with the input value $n = 4$ and $n = 3$.

evaluate ⟦*fibDnC (n)* = E⟧$s\,[n \mapsto 4]$ =
`give the value bound to`
`closure of`
`abstraction of`
evaluate⟦E⟧$s\,[n \mapsto 4]$ =

`give the value bound to`
`closure of`
`abstraction of`
`give the number 4`
`and then`
`give the` $TruthValue$ `of`
`(the given number is equal`
`to the number` 0
`or`
`the given number is equal`
`to the number` 1)
`then`
`check not the given` $TruthValue$
`and then`
`give the sum of`
(**evaluate**⟦$fibDnC(n)$⟧$s\,[n \mapsto 3]$
`and`
evaluate⟦$fibDnC(n)$⟧$s\,[n \mapsto 2]$)

In next step we evaluate the function with the input value $n = 3$ and $n = 2$.

evaluate ⟦*fibDnC (n)* = E⟧$s\,[n \mapsto 3]$ =
`give the value bound to`
`closure of`
`abstraction of`
evaluate⟦E⟧$s\,[n \mapsto 3]$ =

```
give the value bound to
 closure of
  abstraction of
   give the number 3
     and then
   give the TruthValue of
   (the given number is equal
       to the number 0
     or
    the given number is equal
      to the number 1)
   then
    check not the given TruthValue
       and then
        give the sum of
         (evaluate⟦fibDnC(n)⟧s [n ↦ 2]
           and
          evaluate⟦fibDnC(n)⟧s [n ↦ 1])
```

Next step is the evaluation of the function with the input value $n = 2$ and $n = 1$.

evaluate ⟦*fibDnC (n)* = E⟧s [$n \mapsto 2$] =

```
give the value bound to
 closure of
  abstraction of
   evaluate⟦E⟧s [n ↦ 2] =
```

```
give the value bound to
 closure of
  abstraction of
   give the number 2
     and then
   give the TruthValue of
   (the given number is equal
      to the number 0
     or
    the given number is equal
      to the number 1)
   then
    check not the given TruthValue
       and then
        give the sum of
         (evaluate⟦fibDnC(n)⟧s [n ↦ 1]
           and
          evaluate⟦fibDnC(n)⟧s [n ↦ 0])
```

Next we evaluate the function with the input value $n = 1$ and $n = 0$.

evaluate ⟦*fibDnC (n)* = E⟧s [$n \mapsto 1$] =

```
give the value bound to
 closure of
  abstraction of
   evaluate⟦E⟧s [n ↦ 1] =
```

```
give the value bound to
 closure of
  abstraction of
   give the number 1
     and then
   give the TruthValue of
   (the given number is equal
      to the number 0
     or
    the given number is equal
      to the number 1)
   then
    check the given TruthValue
       and then give the number 1
```

evaluate ⟦*fibDnC (n)* = E⟧s [$n \mapsto 0$] =

```
give the value bound to
 closure of
  abstraction of
   evaluate⟦E⟧s [n ↦ 0] =
give the value bound to
 closure of
  abstraction of
   give the number 0
     and then
   give the TruthValue of
   (the given number is equal
      to the number 0
     or
    the given number is equal
      to the number 1)
   then check the given TruthValue
       and then give the number 1
```

After evaluation of the given terms we obtain the final step of description where the function $fibDnC(n)$ is called with the value $n = 0$.

evaluate ⟦*fibDnC (n)* = E⟧s [$n \mapsto 0$] =

```
give the value bound to
 closure of
  abstraction of
   evaluate⟦E⟧s [n ↦ 0] =
```

```
give the value bound to
 closure of
  abstraction of
   give the number 0
     and then
   give the TruthValue of
   (the given number is equal
      to the number 0
     or
    the given number is equal
      to the number 1)
   then check the given TruthValue
       and then give the number 1
```

In the following description we replaced the clausula `the given number` with the concrete values by reason of show-

ing partial results in the evaluation.

```
give the value bound to
 closure of
  abstraction of
        give the sum of
         (the number 1 and the number 1)
      and then
        give the sum of
         (the number 2 and the number 1)
      and then
        give the sum of
         (the number 3 and the number 2)
      and then
        give the sum of
         (the number 5 and the number 3) =
give the value bound to
 closure of
  abstraction of
      give the number 8
```

The result of the function $fibDnC(n)$ for the input value $n = 5$ is equal to 8.

V. Conclusion

Action semantics has a lot of advantages. Descriptions of action semantics offer modularity, modifiability, understandability and readability. The informal appearance and suggestive words of action notation should encourage programmers to read it. A more cryptic notation might discourage programmers from reading it altogether.

Action semantics has a great importance for informatics. It was successfully used in imperative paradigm. In our paper we shown an application of action semantics in functional paradigm and we formulated example of well-known algorithm of the Fibonacci numbers. The next goal of our research will be the extension of action semantics to object-oriented features of OCaml.

Acknowledgement

This work has been supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10: Modelling, simulation and implementation of GPGPU-enabled architectures of high-throughput network security tools.

References

[1] Cheng, F. Mda implementation based on patterns and action semantics. In *2010 Third International Conference on Information and Computing* (2010), vol. 2, pp. 25–28.

[2] Fernandez, M. *Models of Computation. An Introduction to Computability Theory.* Springer-Verlag London Limited, 2009. ISBN 978-1-84882-433-1.

[3] Koubková, A., and Pavelka, J. *Introduction to theoretical informatics.* MatFyzPress, Charles University, Prague, 2005. (in Czech).

[4] Leroy, X. The objective caml system release 3.12. documentation and user's manual. Tech. rep., Institut National de Recherche en Informatique et en Automatique, 2008.

[5] Matoušek, J., and Nešetřil, J. *Kapitoly z diskrétní matematiky*. Nakladatelství Karolinum, Praha, Univerzita Karlova v Praze, 2000.

[6] Mihályi, D., and Novitzká, V. A coalgebra as an intrusion detection system. *Acta Polytechnica Hungarica, Journal of Applied Sciences 7*, 2 (2010). Óbuda University, ISSN 1785-8860.

[7] Mosses, P. D. Theory and practice of action semantics. In *In MFCS '96, Proc. 21st Int. Symp. on Mathematical Foundations of Computer Science* (1996), Springer-Verlag, pp. 37–61.

[8] Nielson, H. R., and Nielson, F. *Semantics with Applications: A Formal Introduction*. John Wiley & Sons, Inc., 2003.

[9] Novitzká, V. *Semantics of programs*. elfa, Košice, 2001. (in Slovak).

[10] Pažin, O. The principles of action semantics. Master's thesis, Technical University of Košice, 2009. (in Slovak).

[11] Pažin, O. The application of action semantics. Master's thesis, Technical University of Košice, 2011. (in Slovak).

[12] Planas, E., Cabot, J., and Gómez, C. Verifying action semantics specifications in uml behavioral models. In *Proceedings of the 21st International Conference on Advanced Information Systems Engineering* (Berlin, Heidelberg, 2009), CAiSE '09, Springer-Verlag, pp. 125–140.

[13] Ruei, R., and Slonneger, K. Semantic prototyping: Implementing action semantics in standard ML. The University of Iowa, 1993.

[14] Sedláček, J. *Úvod do teorie grafů*. Academia, Praha, 1981.

[15] Slodičák, V. Some useful structures for categorical approach for program behavior. *Journal of Information and Organizational Sciences Vol. 35*, No. 1 (2011).

[16] Slodičák, V., and Macko, P. How to apply linear logic in coalgebraical approach of computing. In *Proceedings of the 22nd Central European Conference on Information and Intelligent Systems, September 21-23, 2011, Varaždin, Croatia* (2011), University of Zagreb, pp. 373–380. ISSN 1847-2001.

[17] Slodičák, V., and Macko, P. New approaches in functional programming using algebras and coalgebras. In *European Joint Conferrences on Theory and Practise of Software - ETAPS 2011* (March 2011), Workshop on Generative Technologies, Universität des Saarlandes, Saarbrücken, Germany, pp. pp. 13–23. ISBN 978-963-284-188-5.

[18] Slodičák, V., and Szabó, C. Recursive coalgebras in mathematical theory of programming. In *Proceedings of the 8th Joint Conference on Mathematics and Computer Science* (2010), János Selye University, Komárno. ISBN 978-80-8122-003-6.

[19] Slonneger, K., and Kurz, B. L. *Formal Syntax and Semantics of Programming Languages: A Laboratory Based Approach*. Addison Wesley Longman, 1994.

[20] Stuurman, G. Action semantics applied to model driven engineering, November 2010.

Zusammenfassung

In diesem Artikel stellen wir die Grundlagen der Semantik von Aktionen vor. Wir erwähnen ihre Grundprinzipien und definieren samantische Entitäten. Wir zeigen die Verwendung der Semantik von Aktionen für das funktionale Paradigma, also für die objekt - funktionale Sprache OCaml. Die erreichten Ergebnisse zeigen wir an einem praktischen Beispiel in Informatik. Die Aufzählung von Fibonacci - Zahlen tragen wir in der Sprache OCaml ein und zeigen die Beschreibung der Auswertung in der Semantik von Aktionen.

Satisfiability Problem in Composition-Nominative Logics

Mykola S. Nikitchenko and Valentyn G. Tymofieiev
Department of Theory and Technology of Programming
Taras Shevchenko National University of Kyiv
64, Volodymyrska Street, 01033 Kyiv, Ukraine
nikitchenko@unicyb.kiev.ua, tvuniv@gmail.com

***Abstract*—Composition-nominative logics are algebra-based logics of partial predicates constructed in a semantic-syntactic style. In the paper we present and investigate algorithms for solving satisfiability problem in various classes of composition-nominative logics. This problem is important for verification of specifications presented in logical languages. We consider satisfiability problem for logics of the propositional, renominative, and quantifier levels.**

Compostion-nominative logics; partial predicates; satisfiability; validity

I. INTRODUCTION

The satisfiability problem is a one of the classical problems in logic. In the later time the interest to this problem has raised due to practical value it has obtained in such areas as verification, synthesis, program analysis, testing etc. In this paper we address the satisfiability problem in the context of *composition-nominative approach* [1], which aims to construct a hierarchy of logics of various abstraction levels and generality on the methodological basis, which is common with programming. This approach is based on the principles of *compositionality* and *nominativity*, which are specializations of the well-known principle of development from abstract to concrete.

The analysis made in [1] permits to say that such new logics should be grounded on partial predicates over arbitrary (possibly hierarchical) classes of *nominative data*. Such data consist of pairs *name-value*. Partial predicates over nominative data with names from a set *V* are called here *quasiary*. New predicates are constructed from the basic ones with the help of special operations called *compositions*. According to composition-nominative approach the proposed logics — *composition-nominative logics* (CNL) [2] — are built in a semantic-syntactic style. These logics are based on algebras of partial predicates, which form semantic base for these logics. Such algebras can be defined on different abstraction and generality levels. Classes of terms of such algebras may be considered as sets of formulas of corresponding logics. For such logics axiomatic calculi were constructed and their properties were investigated [2]. On the next step of the development of composition-nominative logics we consider the satisfiability problem for these logics. We study this problem on various levels (propositional level, renominative level, and quantifier level). This is the first paper where we directly address the satisfiability problems in CNL. We aim to reduce these problems to traditional ones in order to apply existent methods for solving them.

II. COMPOSITION-NOMINATIVE LOGICS

When constructing CNL we consider several abstraction levels. These include *propositional*, *singular*, and *nominative*. The level of abstraction defines how data are treated. Algebras of partial predicates (APP) are semantic base for CNL. On different abstraction level we introduce different APP.

Let $Bool=\{F,T\}$ be a set of logical values, $Pr\subseteq Dt\to Bool$ be a subset of partial predicates specified on some data set *Dt*, *C* be a set of *n*-ary compositions of the type $Pr^n\to Pr$. Then a pair (Pr, C) is *an algebra of partial predicates*. The set *Dt* is interpreted as a possible world, elements of *Dt* are treated as world states, predicates from *Pr* are considered as world state properties, and compositions from *C* are treated as operators which construct new predicates from basic ones.

To define compositions we will use the following notations ($P,Q\in Pr, d\in Dt$):

- $P(d)\downarrow= b$ means that predicate *P* on *d* is defined with a value *b*;
- $P(d)\uparrow$ means that predicate *P* on *d* is undefined.

Other undefined notations and terms are understood in the meaning of [2]. On the propositional level data are considered in the most abstract way as "black" boxes. Basic compositions are Kleene's disjunction $\vee$ and negation $\neg$, defined by the following formulas:

$$(P\vee Q)(d)=\begin{cases} T, & \text{if } P(d)\downarrow= T \text{ or } Q(d)\downarrow= T,\\ F, & \text{if } P(d)\downarrow= F \text{ and } Q(d)\downarrow= F,\\ & \text{undefined in other cases.}\end{cases}$$

$$(\neg P)(d)=\begin{cases} T, & \text{if } P(d)\downarrow= F,\\ F, & \text{if } P(d)\downarrow= T,\\ \text{undefined if } & P(d)\uparrow.\end{cases}$$

On the singular level data are considered as concrete "white" boxes. In this case we get one fixed (single) class of data. The corresponding algebras are called *singular*.

Nominative level combines two first levels. Data are considered as "gray" boxes constructed of "white" and "black" boxes. Such data are called *nominative*. They are defined using a set of *names* V and a set of *basic values* A. In this paper we restrict ourselves to nominative data of rank 1, which are partial functions from V into A. Their class is denoted as ${}^{V}A$. We call such data *named sets* (or nominative sets) and usually present them in the form $[v_1 \mapsto a_1,...,v_n \mapsto a_n,...]$, where $v_1,\ldots, v_n,\ldots \in V$, $a_1,\ldots, a_n,... \in A$. If d contains a pair with the name v and the value a, we write $v \mapsto a \in_n d$ or either $a = d(v)$. Predicates and functions over named sets are called quasiary. Their classes are denoted as Pr^A and Fn^A respectively. So, on the nominative level we concretize Dt as ${}^{V}A$.

The previous levels define comparatively simple algebras; in contrast to them the nominative level is rich and can be decomposed into a number of sublevels, in particular, renominative and quantifier levels.

Renominative level

In algebras of this level renaming of data components is allowed. Additional composition is an unary parametric composition of *renomination* $R^{\overline{v}}_{\overline{x}} : Pr^A \to Pr^A$ (here $\overline{v} = (v_1,...,v_n)$ is a list of distinct variables, $\overline{x} = (x_1,...,x_n)$, v_i, $x_i \in V$, $i \in \{1,...,n\}$), which is defined by the following formula:

$$R^{v_1,...,v_n}_{x_1,...,x_n}(P)(d) = P([v \mapsto a \in_n d \mid v \notin \{v_1,...,v_n\}] \nabla [v_i \mapsto a_i \mid x_i \mapsto a_i \in_n d, i \in \{1,...,n\}]).$$

Here $P \in Pr^A$, $d \in {}^{V}A$; we write $v \mapsto a \in_n d$ to denote that a name-value pair with the name v and the value a belongs to d. The ∇ operation is defined as follows. If d_1 and d_2 are two named sets, $d = d_1 \nabla d_2$, then a pair $v \mapsto a \in_n d$ if and only if either $v \mapsto a \in_n d_2$ or $v \mapsto a \in_n d_1$ and $v \mapsto b \notin_n d_2$ for any $b \in A$. For simplicity's sake we will write $d \nabla x \mapsto a$ instead of $d \nabla [x \mapsto a]$.

Quantifier level

On this level predicates can be applied to all data obtained from given data by changing the basic values of the component with a fixed name. Basic compositions are $\vee$, $\neg$, $R^{\overline{v}}_{\overline{x}}$, and existential quantification $\exists x$. The last composition with the parameter $x \in V$ is defined by the formula

$$(\exists x P)(d) = \begin{cases} T, & \text{if } b \in A \text{ exists}: P(d \nabla x \mapsto b) \downarrow = T, \\ F, & P(d \nabla x \mapsto a) \downarrow = F \text{ for each } a \in A, \\ & \text{undefined in other cases.} \end{cases}$$

Derived compositions (such as conjunction &, universal quantification $\forall x$, etc.) are defined in a traditional way.

An important property of CNL compositions considered here is monotonicity. We say that $P \subseteq Q$, where $P, Q \in Pr$ if for any $d \in Dt$ such that $P(d) \downarrow = b$ we have $Q(d) \downarrow = b$. We say that $(P_1,\ldots,P_n) \subseteq (Q_1,\ldots,Q_n)$ if $P_i \subseteq Q_i$ for any $i \in \{1,...,n\}$. An n-ary composition $c : Pr \times \ldots \times Pr \to Pr$ is called *monotone* iff from $(P_1,\ldots,P_n) \subseteq (Q_1,\ldots,Q_n)$ follows $c(P_1,\ldots,P_n) \subseteq c(Q_1,\ldots,Q_n)$. It it is easy to prove that $\vee$, $\neg$, renomination $R^{\overline{v}}_{\overline{x}}$, and existential quantification $\exists x$ are monotone compositions. For example, let us show monotonicity of $\exists x$. Assume $P \subseteq Q$. Let $\exists x\, P(d) \downarrow = T$. It means that there exists $b \in A$ such that $P(d \nabla x \mapsto b) \downarrow = T$; thus we have that $Q(d \nabla x \mapsto b) \downarrow = T$; therefore $\exists x\, Q(d) \downarrow = T$. Let $\exists x\, P(d) \downarrow = F$. It means that for any $a \in A$ $P(d \nabla x \mapsto a) \downarrow = F$; thus we have that for any $a \in A$ $Q(d \nabla x \mapsto a) \downarrow = F$. This means that $\exists x\, Q(d) \downarrow = F$.

On each abstraction level various classes of predicates can be specified. For the nominative level the most important are classes of *equitone* (denoted by EPr^A) and *maxitotal equitone* predicates. Equitone predicates preserve their values under data extensions; maxitotal equitone predicates additionally should be defined on maximal extensions of data. More detailed description of these classes of predicates and their properties can be found in [1, 2].

Summing up, we can say that composition-nominative logics are based on different algebras of partial predicates.

In the sequel we consider such levels of CNL: propositional, renominative, and quantifier level.

Now we describe the language of pure (without functional symbols) CNL. Let Ps be an arbitrary set called the set of predicate symbols. Let C be the set of basic composition names. The set C is defined by the level of CNL as described above. Alphabet of CNL consists of symbols from sets Ps, C, and V. Main constructions of CNL are formulas. Formulas (terms of corresponding algebras) are built inductively as follows:

- each $p \in Ps$ is an atomic formula
- if $\Phi_1,\ldots, \Phi_n$ are formulas, $c \in C$ is an n-ary composition then $c\,\Phi_1\ldots\Phi_n$ is a formula.

The only uninterpreted symbols in formulas of CNL are predicate symbols. An *interpretation* of a CNL formula is a triple $J = (A, Pr, I)$, where Pr is a class of predicates on A, I is a total function $I: Ps \to Pr$. The pair (Pr, C) is a (sub)algebra. That is why we say that algebras of partial predicates form the semantic base of the CNL. We concretize Pr as Pr^A in the case of renominative and quantifier levels of CNL, which are induced by the nominative data abstraction level. An interpretation $J = (A, Pr, I)$ is said to be *total* if I maps each $p \in Ps$ to a total predicate $I(p) \in Pr^A$.

Given an interpretation J each CNL formula Φ is interpreted as a predicate Φ_J. If Φ is an atomic formula then $\Phi_J = I(\Phi)$, otherwise it is defined with respect to semantics of correspondent composition operations.

In the sequel we consider formulas in their traditional form using infix operations and brackets; brackets can be omitted according usual priorities rules.

III. SATISFIABILITY PROBLEM FOR CNL

A CNL formula Φ is called *satisfiable on an interpretation* $J = (Dt, Pr, I)$ if there is a $d \in Dt$ such that $\Phi_J(d)\downarrow = T$. We shall denote this as $J \models \Phi$. A CNL formula Φ is called *satisfiable in a predicate class Pr* if there exists an interpretation $J = (Dt, Pr, I)$ on which Φ is satisfiable. We shall denote this as $(Dt, Pr) \models \Phi$. A CNL formula Φ is called *satisfiable* if there exists an interpretation J on which Φ is satisfiable. We shall denote this as $\models \Phi$.

Satisfiability of a formula is related to its validity. A CNL formula Φ is called *valid on an interpretation* $J = (Dt, Pr, I)$ if there is no $d \in Dt$ such that $\Phi_J(d)\downarrow = F$. We shall denote this as $J \models \Phi$. A CNL formula Φ is called *valid in predicate class Pr* if for any interpretation $J = (Dt, Pr, I)$ we have $J \models \Phi$. We shall denote this as $(Dt, Pr) \models \Phi$. Formula Φ is called *valid* if $J \models \Phi$ for any interpretation J. Due to possible presence of partial predicates we do not have the property that Φ is valid iff $\neg\Phi$ is not satisfiable, which holds for classical first-order logics. We call formulas Φ and Ψ *equisatisfiable* if they are either both satisfiable or both not satisfiable (i.e., unsatisfiable).

Let *TPr* be the subset of all total predicates from *Pr*, i.e., predicates that are defined on all data from the correspondent domain.

Theorem 1. Let $Pr \subseteq Dt \to Bool$ be a class of partial predicates, (Pr, C) be a predicate algebra, Φ be a term (formula) of this algebra. Suppose that all compositions in C are monotone. Then $(Dt, Pr) \models \Phi$ iff $(Dt, TPr) \models \Phi$.

Proof. To prove this theorem it is sufficient to show that for every interpretation $J = (Dt, Pr, I_1)$ such that $J \models \Phi$ we can construct an interpretation $K = (Dt, TPr, I_2)$ such that $K \models \Phi$. Let *Ps* be the set of predicate symbols occurring in Φ. For each $p \in Pr$ we define a predicate $ext(p)$ such that for each $d \in Dt$ $ext(p)(d) = p(d)$ if $p(d)$ is defined and $ext(p)(d) = T$ if $p(d)$ is undefined. Note that $ext(p) \in TPr$ and $p \subseteq ext(p)$. For each $p \in Ps$ we now put $I_2(p) = ext(I_1(p))$. Due to monotonicity of compositions it follows easily that $\Phi_J \subseteq \Phi_K$. From $J \models \Phi$ it now follows $K \models \Phi$. ♦

Theorem 1 justifies that it is sufficient to consider only total interpretations in order to check satisfiability of a CNL formula Φ. We will use this theorem to study satisfiability primarily on the propositional level; for nominative levels it is usually not convenient to consider total predicates.

A. Propositional abstraction level

Let $TCPr \subset Pr$ be a set of total constant predicates $TCPr = \{True, False\}$ such that for each $d \in Dt$ $True(d)\downarrow = T$, $False(d)\downarrow = F$.

Theorem 2. Let Φ be a propositional CNL formula. Then $(Dt, TPr) \models \Phi$ iff $(Dt, TCPr) \models \Phi$.

Proof. To prove this theorem it is sufficient to show that for every interpretation $J = (Dt, TPr, I_1)$ such that $J \models \Phi$ we can construct an interpretation $K = (Dt, TCPr, I_2)$ such that $K \models \Phi$. Let *Ps* be the set of predicate symbols occurring in Φ. From $J \models \Phi$ it follows that there exist $d \in Dt$ such that $\Phi_J(d)\downarrow = T$. For every $p \in Ps$ let $I_2(p)$ be a total constant predicate taking the value $I_1(p)(d)$. It is easily shown that $\Phi_K(d) \downarrow = T$, which means $K \models \Phi$. ♦

Theorem 2 justifies considering only total constant interpretations for checking satisfiability of propositional CNL formulas. In fact, such reductions can be continued. On the propositional level of CNL we can consider data on singular abstraction level.

Theorem 3. Let $d \in Dt$. Then $(Dt, TCPr) \models \Phi$ iff $(\{d\}, TCPr) \models \Phi$.

The proof is trivial.

Summarizing theorems 1–3 we obtain such chain of reductions:

$$(Dt, Pr) \models \Phi \Leftrightarrow (Dt, TPr) \models \Phi \Leftrightarrow (Dt, TCPr) \models \Phi \Leftrightarrow$$

$$\Leftrightarrow (\{a\}, TCPr) \models \Phi.$$

On the last element of the chain we assign to each predicate symbol in Φ a constant true/false predicate defined on a singleton set. This essentially means that for checking $\models \Phi$ on the propositional level we can use the methods for checking satisfiability in the classical propositional logic. Among possible approaches to handling propositional satisfiability we would like to mention Davis-Putnam-Logemann-Loveland algorithm [3], variations of which are very widely used.

B. Nominative abstraction level

On the nominative level of data treating we consider partial predicates over named sets as the most general class of predicates. Thus, on this level when concretizing the data set *Dt* we refer to $\models \Phi$ as $({}^{V}A, Pr^{A}) \models \Phi$.

Renominative level of CNL

On this level we suggest a technique that allows reducing the satisfiability problem for CNL formula of the renominative level (RCNL formula) to the satisfiability problem for classical predicate logic. An RCNL formula is said to be in *renominative normal form* (RNF) if the renomination composition is applied only to predicate symbols. It means that for any subformula of the form $R_{\bar{x}}^{\bar{y}}(\Psi)$ we have that $\Psi \in Ps$. We then call $R_{\bar{x}}^{\bar{y}}(P)$ a *renominative atom*. To construct the normal form of an arbitrary RCNL formula we apply the following transformations from top to bottom according to formula structure:

1) $R_{\bar{x}}^{\bar{y}}(\neg P) \Rightarrow \neg R_{\bar{x}}^{\bar{y}}(P)$

2) $R_{\bar{x}}^{\bar{y}}(P \vee Q) \Rightarrow R_{\bar{x}}^{\bar{y}}(P) \vee R_{\bar{x}}^{\bar{y}}(Q)$.

3) $R^{v_1,\ldots,v_n,w_1,\ldots,w_m}_{x_1,\ldots,x_n,y_1,\ldots,y_m}(R^{v_1,\ldots,v_n,u_1,\ldots,u_k}_{s_1,\ldots,s_n,z_1,\ldots,z_k}(P)) \Rightarrow$
$R^{v_1,\ldots,v_n,w_1,\ldots,w_m,u_1,\ldots,u_k}_{\alpha_1,\ldots,\alpha_n,y_1,\ldots,y_m,\beta_1,\ldots,\beta_k}(P)$.

Here $\alpha_i = s_i(v_1,\ldots,v_n,w_1,\ldots,w_m \;/\; x_1,\ldots,x_n,\; y_1,\ldots,y_m)$, $\beta_j = z_j(v_1,\ldots,v_n,\; w_1,\ldots,w_m \;/\; x_1,\ldots,x_n,\; y_1,\ldots,y_m)$, where $r(b_1,\ldots,b_q \;/\; c_1,\ldots,c_q) = r$ if $r \notin \{b_1,\ldots,b_q\}$, $r(b_1,\ldots,b_q \;/\; c_1,\ldots,c_q) = c_i$ if $r = b_i$ for some i.

It can easily be checked that by applying transformations 1–3 we obtain an equisatisfiable formula.

Note that mentioned transformations preserve propositional structure of the formula in the sense that they do not affect the reciprocal order of other compositions in the formula structure except to renomination.

Let now Φ be an arbitrary RCNL formula in renominative normal form, *Ps* be the set of predicate symbols occurring in Φ. Next step is unifying renominative atoms with the same predicate symbols. Let $P \in Ps$, $R^{\overline{v}}_{\overline{x}}(P)$, $R^{\overline{w}}_{\overline{y}}(P)$ be two different occurrences of renominative atoms with P in Φ. We generalize vectors $\overline{v}$ and $\overline{w}$ in order to obtain the occurrences of the form $R^{\overline{u}}_{\overline{q}}(P)$, $R^{\overline{u}}_{\overline{r}}(P)$. This is done with the help of the following rule: $R^{\overline{v}}_{\overline{x}}(P) = R^{z,\overline{v}}_{z,\overline{x}}(P)$, $z \neq v_i$. Occurrences of P without the renomination operation are substituted with $R^{\overline{u}}_{\overline{u}}(P)$. Nothing is performed in the case when all occurrences of P are without $R^{\overline{v}}_{\overline{x}}$ composition. It is easy to see that this transformation produces an equisatisfiable formula. Now we construct an equisatisfiable formula Ψ of classical predicate logic. Satisfiability of formulas in this logic is understood in the usual way. Formula Ψ is obtained from Φ by replacing each $R^{v_1,\ldots,v_n}_{x_1,\ldots,x_n}(P)$ occurrence by an n-ary predicate $P(x_1,\ldots,x_n)$. Thus, the set of predicate symbols in Ψ is same as of Φ. Predicates occurring without renomination in Φ are replaced by 0-ary predicates in Ψ.

Theorem 4. For any RCNL formula Φ there can be effectively constructed a quantifier-free formula Ψ of classical predicate logic such that Φ and Ψ are equisatisfiable.

Proof. It is sufficient to show that Φ and Ψ are equisatisfiable, where Ψ is a formula constructed in the above described manner. Let $J = ({}^{V}A, TPr^{A}, I)$ be an interpretation, $a \in {}^{V}A$ such that $\Phi_J(a)\downarrow = T$. Considering only total interpretations is justified by theorem 1. Let K be an interpretation function for predicate symbols in Ψ, which assigns to each n-ary predicate symbol P some n-ary relation over A. For each renominative atom $R^{v_1,\ldots,v_n}_{x_1,\ldots,x_n}(P)$ occurring in the RNF of Φ we assign

$$K(P)(a(x_1),\ldots,a(x_n)) = (R^{v_1,\ldots,v_n}_{x_1,\ldots,x_n}(P))_J(a).$$

For other tuples $K(P)$ can be assigned arbitrarily. It is easy to see that with such interpretation of predicates in Ψ we have that Ψ is satisfiable as well, namely $\Psi(a)\downarrow = T$. Here by $\Psi(a)$ we denote a substitution of the correspondent values of the data a into arguments of the predicates in Ψ. Say, for instance, the occurrence of a predicate P in Ψ corresponds to the occurrence of $R^{v_1,\ldots,v_n}_{x_1,\ldots,x_n}(P)$ in the formula Φ; in such case $a(x_1),\ldots,a(x_n)$ are the arguments of P in $\Psi(a)$. Similarly one can show that satisfiability of Φ in RCNL follows from satisfiability of Ψ in predicate logic.

This allows us to use methods for checking the satisfiability problem in classical predicate logic. Formulas of this logic can further be transformed into equisatifiable formulas of pure propositional logic. However this requires the data set A have enough number of values to distinguish all $(x_1,\ldots,x_n)$ tuples occurring in Φ. Anyhow it is not important if we consider general satisfiability problem, i.e., when we do not concretize the data set. Thus, on the renominative level we can reduce the CNL satisfiability problem to the propositional satisfiability problem. The reduction is illustrated on the following examples. For convenience, we use infix notation for CNL formulas along with round brackets to embrace arguments of operations when needed.

Example 1. Consider an RCNL formula Φ with three basic predicates: *P, Q, S.* Let the set V contain x_1, y_1, x_2, y_2. Let

$$\Phi = R^{x_1}_{y_1}(P \vee Q) \,\&\, \neg R^{x_1,x_2}_{y_1,y_2}(P \vee R^{x_2,y_1}_{x_1,y_2}Q) \,\&\, S.$$

According to procedure, we construct renominative normal form of Φ by applying simple transformations. Thus we obtain Φ_1.

$$\Phi_1 = (R^{x_1}_{y_1}P \vee R^{x_1}_{y_1}Q) \,\&\, \neg(R^{x_1,x_2}_{y_1,y_2}P \vee R^{x_1,x_2,y_1}_{y_1,y_1,y_2}Q) \,\&\, S.$$

Now we unify occurrences of the renominative atoms $R^{x_1}_{y_1}P$ and $R^{x_1,x_2}_{y_1,y_2}P$ along with $R^{x_1}_{y_1}Q$ and $R^{x_1,x_2,y_1}_{y_1,y_1,y_2}Q$. Thus we obtain Φ_2.

$$\Phi_2 = (R^{x_1,x_2}_{y_1,y_2}P \vee R^{x_1,x_2,y_1}_{y_1,x_2,y_1}Q) \,\&\, \neg(R^{x_1,x_2}_{y_1,y_2}P \vee \; R^{x_1,x_2,y_1}_{y_1,y_1,y_2}Q) \,\&\, S.$$

Now we are ready to construct a predicate logic formula Ψ having a 0-ary predicate s, binary predicate p and ternary predicate q.

$$\Psi = (p(y_1, x_2) \vee q(y_1, x_2, y_1)) \,\&\, \neg(p(y_1,y_2) \vee q(y_1,y_1,y_2)) \,\&\, s.$$

This formula is satisfiable. For example, we can consider the set of natural numbers as a data set A. Then the latter formula is equisatisfiable to the following formula of propositional logic $(p_1 \vee q_1) \,\&\, \neg\, (\, p_2 \vee q_2) \,\&\, s$, which is clearly satisfiable. Thus we've established $|\approx \Phi$. Note that we do not have $(V, \{a\}) |\approx \Phi$. Indeed, if we use one-element data set then Ψ would be equisatisfiable with the propositional formula $(p \vee q) \,\&\, \neg(p \vee q) \,\&\, s$, which is not satisfiable.

Example 2. Consider an RCNL formula Φ with two basic predicates: *P, Q*. Let the set *V* contains x_1, y_1, x_2, y_2.

$$\Phi = R^{x_1,x_2}_{y_1,y_2}(P \,\&\, Q) \,\&\, \neg R^{x_2}_{y_2}(R^{x_1}_{y_1} P)\,.$$

RNF of Φ is then

$$\Phi_1 = R^{x_1,x_2}_{y_1,y_2} P \,\&\, R^{x_1,x_2}_{y_1,y_2} Q \,\&\, \neg R^{x_1,x_2}_{y_1,y_2} P\,.$$

No unification of renominative atoms needed, so an equisatisfiable predicate logic formula Ψ has two binary predicates *p* and *q* and is as follows:

$$\Psi = p(y_1, y_2) \,\&\, q(y_1, y_2) \,\&\, \neg p(y_1, y_2).$$

This formula is unsatisfiable, which means that Φ is unsatisfiable as well.

Quantifier level of CNL

Due to Church, the satisfiability problem for the first order predicate logic (FOPL) is undecidable. For every FOPL formula we can construct an equisatisfiable CNL formula of the quantifier level (QCNL formula). That is why on the quantifier level the satisfiability problem for CNL is undecidable in general case. One possible way of construction QCNL formula Φ from a FOPL formula *F* is replacing each occurrence of $p(x_1,..,x_n)$ by $R^{1,\ldots,n}_{x_1,\ldots,x_n}(p(x_1,\ldots,x_n))$, where *p* is a predicate symbol, $x_1,\ldots, x_n$ are variables in *F*, and $1,\ldots, n$ are considered as new names. As to the set of basic values *A*, we take the same as used in *F*, as to the set of names *V*, we take all variables occurring in *F* and extend this set with new variables $\{1,\ldots, k\}$, where *k* is the maximal arity of predicate symbols in *F*. This is how *n*-ary predicates can be specified in QCNL. The satisfiability problem for *F* can now be formulated as checking $({}^{V}A, Pr^{A}) \models \Phi$.

Even though the satisfiability problem for QCNL is undecidable it is still important to be able to solve this problem in some particular cases. We now show that an analogous approach as proposed for handling RCNL formulas is not quite sufficient in QCNL case. Again, let us describe the transformation of the formula Φ to a formula Ψ of FOPL. This reduction is called classical normalization. This normalization is done in two steps. During the first step the renominative normal form of Φ is constructed. The notion of renominative normal form for QCNL is the same as for RCNL. This is done by the same scheme as in the case of RCNL formulas. As to $\exists x$ composition, it is handled by the rule $R^{\bar{v}}_{\bar{x}}(\exists y P) \Rightarrow \exists y R^{\bar{v}}_{\bar{x}}(P)$, when $y \notin \{\bar{v}, \bar{x}\}$. If $y \in \{\bar{v}\}$ and $y \notin \{\bar{x}\}$, then the renomination by *y* can simply be omitted, i.e. we apply the rule $R^{y,\bar{v}}_{z,\bar{x}}(\exists y P) \Rightarrow \exists y R^{\bar{v}}_{\bar{x}}(P)$. In the case when $y \in \bar{x}$ we extend the set of names *V* by introducing a fresh variable λ and apply the transformation $R^{z,\bar{v}}_{y,\bar{x}}(\exists y P) \Rightarrow R^{z,\bar{v}}_{y,\bar{x}}(\exists \lambda\, R^{y}_{\lambda}(P))$. Afterwards the general rules described previously can be applied. Introducing new variables is actually language extension; however it is easy to see that it does not affect the satisfiability of the formula. We can also consider the satisfiability problem for the same formula in an extended language, which has unlimited amount of unused names.

During the second step, having put Φ in normal form, again, formula Ψ is obtained from Φ by replacing each $R^{v_1,\ldots,v_n}_{x_1,\ldots,x_n}(P)$ occurrence by an *n*-ary predicate $P(x_1,\ldots,x_n)$. Predicates occurring in Φ without renomination are replaced by 0-ary predicates in Ψ.

Are the formulas Φ and Ψ equisatisfiable? This is true for many classes on CNL, in particular, for neoclassical CNL based on the notion of equitone predicate. But in general case the equisatisfiability may be lost. This is illustrated with the example below.

Example 3. Consider a CNL formula

$$\Phi = (\forall x \neg P) \,\&\, P\,.$$

It is easy to show that this formula is satisfiable in QCNL. Indeed, let $J = ({}^{V}A, Pr^{A}, I)$ such that $V=\{x,y\}$, $A=\{1,2\}$. Let $I(P)(d)\downarrow = F$ if a pair $x \mapsto a \in_n d$ for some $a \in A$ and *T* in all other cases. In other words, the predicate *P* takes the value *T* on some data *d* iff the name *x* is undefined in *d*. Note that the predicate *I*(*P*) is not equitone. Indeed, $I(P)([\, x \mapsto 1, y \mapsto 1\,])\downarrow = F$, whereas $I(P)([\, y \mapsto 1\,])\downarrow = T$. Now we have that $\Phi_J([\, y \mapsto 1\,])\downarrow = T$, which means that Φ is satisfiable. At the same time it is easy to prove that the formula

$$(\forall x \neg P(x)) \,\&\, P(x)\,,$$

which is a classical normalization of Φ, is not satisfiable in classical first order logic.

This example justifies that in order to check satisfiability of a general CNL formula Φ it is not sufficient to consider only interpretations and methods of classical logics. More elaborated methods should be developed.

The satisfiability problem for the FOPL is often addressed for specialized theories, when some predicates have specified interpretations and several axioms shall hold for admissible interpretations. This is often referred to as satisfiability modulo theory (SMT) problem [4]. For some logical theories, for example, linear integer arithmetic (Presburger arithmetic) this problem is decidable. In most cases for the theories having a practical value, the SMT problem is decidable only for quantifier-free fragments [5]. The SMT problem often occurs in practice in the areas of software/hardware verification, program analysis, testing etc. Therefore we consider the SMT problem for composition-nominative logics as an important instance of the general satisfiability problem.

IV. Conclusions

In this paper we have defined and investigated the satisfiability problem for the composition-nominative logic.

We have considered CNL on three different levels: propositional, renominative, and quantifier level. For these levels we have proposed possible ways of reduction of the satisfiability problem and ways of application of the algorithms already developed. For propositional CNL we have presented a technique for transforming original CNL formula to an equisatifiable formula of pure propositional logic. For CNL of the renominative level we have proposed a method of reduction of the satisfiability problem to the same problem for classical first-order predicate logic. For quantifier level of CNL we have shown that the problem considered is undecidable in general case. We have also outlined that for CNL formulas the class of interpretations on which some formula is satisfiable can be different from that of classical first order logic. This is due to possible presence of non-equitone predicates.

This study is essentially a first step in investigating the satisfiability problem for composition-nominative logic. It has shown possible important directions of further satisfiability procedures development. One of them is investigating and outlining decidable fragments of QCNL. In particular we are interested in solving the problem of satisfiability modulo theories. Other important problem for all kinds of CNL discussed in this paper is constructing algorithms for validity checking.

REFERENCES

[1] N. Nikitchenko, "A composition nominative approach to program semantics," Technical report IT–TR 1998-020, Technical University of Denmark, 1998.

[2] M. Nikitchenko and S. Shkilnyak, Mathematical logic and theory of algorithms. Publishing house of Taras Shevchenko National University of Kyiv, 2008 (in Ukrainian).

[3] M. Davis, G. Logemann, and D. Loveland, "A machine program for theorem-proving," Comm. of the ACM, vol. 5(7), 1962, pp. 394-397.

[4] R. Nieuwenhuis, A.Oliveras, and C.Tinelli, "Solving SAT and SAT modulo theories: from an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T)," Journal of the ACM, 53, 2006, pp. 937-977.

[5] D. Kroening and O. Strichman, Decision procedures — an algorithmic point of view. Springer, 2008.

Acquiring Information from Ontologies with OntoP

Ines Čeh, Milan Zorman, Matej Črepinšek, Tomaž Kosar, Marjan Mernik
Faculty of Electrical Engineering and Computer Science
University of Maribor
Maribor, Slovenia
{ines.ceh, milan.zorman, matej.crepinsek, tomaz.kosar, marjan.mernik}@uni-mb.si

Jaroslav Porubän
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
jaroslav.poruban@tuke.sk

***Abstract*—This paper presents a parser for OWL DL. OWL DL, a sublanguage of OWL, allows efficient reasoning with computability assurance. The OWL parser, named OntoP, was developed specifically for the requirements of a larger framework: Ontology2DSL. Ontology2DSL enables the semi-automated construction of a formal grammar as well as programs incorporating Domain-Specific Languages (DSL) from OWL ontologies. In this presentation of the parser, we focus on the data structure that the parser uses to store the extracted information from an OWL document (written in RDF/XML syntax) and the algorithm used to construct and visualize the class hierarchy of the ontology.**

Keywords: parsing; ontology; OWL; RDF/XML syntax; Manchester OWL syntax

I. INTRODUCTION

Web Ontology Language (OWL) is a semantic markup language developed for the representation of information on the semantic web [1], [2]. It was designed by the World Wide Web Consortium (W3C). It reached the status of a recommendation in 2004. OWL ontologies are used for modeling domain knowledge. An OWL ontology is a set of axioms that describe classes, properties and the relations among them. A review of existing literature provides many definitions of an ontology. The most widely accepted is the definition by Studer et al. that defines ontology as: "An ontology is a formal, explicit specification of a shared conceptualization." [3], [4].

Besides its formal W3C recommendation status, OWL is a success because of the vast set of tools that enable one to work with OWL ontologies. Tools enable the creating and editing of OWL ontologies (Protégé [5], Swoop [6]), offer inference and reasoning (Pellet [7], Fact++ [8]). In addition to these tools, there are also various Application Programming Interfaces (APIs) that enable the use of ontologies in various applications (OWL-API [9] and Jena [10]).

OntoP, the parser that we will present in further detail in section 3, was developed for the Ontology2DSL framework [11]. The Ontology2DSL framework, shown as a workflow diagram in Fig. 1, enables the semi-automated construction of formal grammars [12] of DSL's [13] and programs, all from an OWL ontology. The framework accepts an OWL document as its input. It then proceeds with parsing the document and uses the acquired information to fill its internal data structure, known as an ontology data structure (ODS). The transformation pattern (TP), a sequence of rules that construct the grammar and programs, is run over ODS. Some of the rules in the TP require some involvement and activity from a DSL engineer. The final results of the framework are the DSL grammars and programs.

The construction of language grammar and programs from an OWL ontology is executed in four steps. In the first step, the user selects the ontology, over which the transformation will run. In the second step, they select the pattern that will be used in the transformation. Since the transformation usually does not include the entire ontology (DSL requirements do not match fully with all of ontology concepts) the DSL engineer in step three excludes the classes not used in the transformation from the class hierarchy. In step four, the TP is run over the ODS. Step four comprises a variable number of substeps. This number is dependent on the number of rules that the individual TP holds.

The OntoP parser is one of the five main components of the Ontology2DSL framework. The architecture and major components of the framework are presented in more detail in [11]. The tasks of the OntoP parser, in the process of constructing the language grammar and programs, include the following:

- Creating and filling the internal data structure ODS from a specific OWL document. The internal data structure is fully compliant with the input document.
- Creating and visualizing the ontology class hierarchy.
- Acquiring and presenting all available information for each of the classes from the hierarchy.

The Ontology2DSL framework also provides a browsing capability to the DSL engineers interested in particular ontologies. To facilitate this functionality, we have extended the OntoP parser. The parser thus displays the hierarchy of properties and lists of individuals in addition to visualizing the class hierarchy. The parser also provides all of the information

on the ontology, such as the comments written during the ontology's development, versioning information, etc.

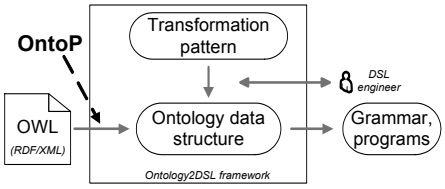


Figure 1. Ontology2DSL framework workflow diagram.

This paper will present the OntoP parser, developed in the programming language C#, ODS, and the algorithm used for the construction and visualization of the class hierarchy.

The organization of the paper is as follows: Section 2 presents the ontology web language OWL. Section 3 introduces the OntoP parser, ODS and the algorithm for construction and visualization of the class hierarchy. The conclusion and ideas for future work are summarized in Section 4.

II. The Web Ontology Language OWL

A. OWL Sublanguages

The original OWL specification includes the definition of three sublanguages of OWL, with different levels of expressiveness. These languages are OWL Full, OWL DL and OWL Lite (ordered by decreasing expressiveness) [1], [2], [14]. OWL Full is not a sublanguage of the OWL language, it is the full OWL language. OWL Full allows the user to "say anything about anything." The flexibility of OWL Full is detrimental to computational efficiency. OWL Full is not decidable. This means that there are no known algorithms that are capable of assuring complete inference for a given OWL Full ontology. The "DL" in the name of OWL DL sublanguage stands for Description Logic, an important subset of first-order logic [15]. OWL DL uses the same constructs as OWL Full. The use of some of these constructs is restricted [16]. The restrictions make OWL DL decidable. Therefore, an algorithm exists that ensures complete inference for any OWL DL ontology. Decidability, however, makes no statements on the efficiency of the algorithm and does not guarantee real-time completion. OWL Lite is essentially OWL DL with a subset of its language constructs.

B. OWL DL

The three basic components of the OWL DL sublanguage, which in this paper is only referred to as OWL, are: classes, properties and individuals [1], [5].

OWL defines two types of classes: simple named and predefined classes. Simple named classes are classes defined by the user. Predefined classes, provided by OWL, are "Thing" and "Nothing." While "Thing" is the superclass of all classes, "Nothing" is an empty class and it can be a subclass of every class. Classes can be organized into hierarchies. The OWL language allows multiple inheritance. Inheritance refers to the inheritance of properties inherited by children from parents. Classes can be created with a combination of other classes with the use of intersection, union and complement operators. The second component is properties, which is a binary relation. The two main types of properties in OWL are object properties and datatype properties. Object properties link objects with other objects, datatype properties with data values. OWL uses data values defined in the XML Schema Definition Language (XSD) (version 1.0) [1]. A property can have multiple domains and ranges. Object properties can also have inverse properties. In that case, the domain and range are swapped. OWL allows properties to have special characteristics. Consequently, properties can be transitive, symmetrical, functional and inverse functional. Object and datatype properties can be organized into hierarchies. The third component is individuals, which are members of user-defined classes.

A document is an OWL DL if its corresponding graph conforms to the rules for OWL-DL [17].

C. Syntaxes

A syntax is a set of rules that define the language format. The OWL W3C recommendation defines the standard exchange syntax as the RDF/XML syntax [1], [2], [15]. This is the syntax that all OWL compatible tools should support. Besides the RDF/XML syntax, other syntaxes also exist. Some examples include: Turtle, Abstract Syntax [1], [2], Manchester OWL [18] and others. Different syntaxes are optimized for different purposes. For our work, the most important syntaxes are the RDF/XML and the Manchester OWL syntax.

RDF/OWL, the primary syntax that all OWL compatible tools must support, is the XML syntax intended to represent RDF triples. RDF/XML is a very extensive syntax. Most tools use this syntax as the default for saving OWL ontologies. The advantage of this syntax is that it is widely supported. The disadvantage is that it is very extensive and difficult for humans to comprehend.

Manchester OWL, the compact text syntax, is derived from Abstract Syntax. Compared to it, Manchester OWL is less extensive and minimizes the use of brackets. This results in it being easy to read, write and edit. Relatively difficult expressions written in this syntax should be as easily readable as regular English language. Syntax uses natural language words such as "AND", "SOME" and "NOT" instead of mathematical symbols. Reading and understanding is also made easier with the infix notation. The simplicity of the syntax is the major advantage. The disadvantage is that it is very clumsy for some OWL axioms.

Fig. 2 shows how compact and readable the Manchester syntax is when compared to XML/RDF syntax by showing the definition of VegetarianPizza written in RDF/XML (Fig. 2a) and in Manchester OWL (Fig. 2b) syntax.

```
<owl:Class rdf:about="#VegetarianPizza">
<owl:equivalentClass>
  <owl:Class>
    <owl:intersectionOf rdf:parseType="Collection">
     <rdf:Description rdf:about="#Pizza"/>
     <owl:Restriction>
       <owl:onProperty rdf:resource="#hasTopping"/>
        <owl:allValuesFrom>
         <owl:Class>
           <owl:unionOf rdf:parseType="Collection">
             <rdf:Description rdf:about="#CheeseTopping"/>
             <rdf:Description rdf:about="#VegetableTopping"/>
           </owl:unionOf>
         </owl:Class>
       </owl:allValuesFrom>
     </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
</owl:equivalentClass>
</owl:Class>
```

a)

```
Class: VegetarianPizza
Pizza and hasTopping only (CheeseTopping or VegetableTopping)
```

b)

Figure 2. Definition of VegetarianPizza class written in RDF/XML (a) and in Manchester OWL (b) syntax.

III. ONTOP PARSER

Parsing is the procedure in which the parser takes a concrete representation of a document, which contains an ontology, and then builds an internal representation compliant with the document.

OntoP is a parser implemented in the programming language C# which is part of the .NET framework. OntoP uses the XML document object model (DOM) and the XML parsers associated with it to parse RDF/XML documents. It is intended to parse OWL ontologies. OntoP supports all syntactic elements of OWL language. Elements of the language, their descriptions and examples can be found in [1], [16], [19].

As previously mentioned, the OntoP parser is used within the Ontology2DSL framework for two scenarios. The first is when developing DSL from OWL ontologies; the second is the browsing capability it gives to DSL engineers.

Within the first scenario, the parser performs the following tasks:

- It takes a concrete representation of an OWL document written in RDF/XML syntax and constructs the inner representation (fully compliant with the OWL document). The inner representation is the ontology data structure, ODS, to which OntoP writes the extracted information. RDF/XML syntax was selected because it is the default choice in most tools. This choice has allowed us the widest choice of ontologies that can be incorporated for DSL development.
- It retrieves the list of classes contained in the ontology and constructs the appropriate class hierarchy. The ontology hierarchy is visualized in the form of a tree in which each node is a class from the ontology. A visual representation of the ontology diminishes the time needed for DSL development. This is because not all ontology classes are required for DSL development. Unnecessary classes need to be removed. With the visual representation, the DSL engineer can easily deselect classes that he does not require in the transformation and therefore eliminates the long procedure of eliminating unnecessary classes from the RDF/XML syntax.
- The parser then extracts all information for a particular class and displays them to the DSL engineer. Class descriptions are written in Manchester OWL syntax, which was chosen for its easy readability. The visualization of class information is important because the better the ontology is known, the better the resulting DSL can be. Easily readable syntax shortens the development cycle.

When browsing the ontology, the parser also performs the following tasks: it displays the hierarchies of object and datatype properties, the list of individuals and all information on objects and datatype properties and individuals. It also acquires and displays all information on the ontology.

A. Ontology data structure

OntoP fills the ontology data structure with the extracted information. ODS is the data structure used to run transformation patterns over. ODS is composed of the following data structures: tree of classes, tree of object properties, tree of datatype properties, and a list of individuals. The parser fills the ODS in the same sequence as they are listed here. The nodes in individual trees are objects that contain information on the individual ontology block stored in the node. Each node stores the name of the block, as well as the following:

- equivalent classes, superclasses, members, disjoint classes, comments and labels (Class tree).
- characteristics, domains, ranges, inverse properties and super properties (Object property tree).
- characteristics, domains, ranges and super properties (Data type property tree).
- types, data property assertion and object property assertions (Individual tree).

B. Hierarchy construction and visualization algorithm

For the construction of the class hierarchy, the OntoP parser uses a two-part algorithm. The first part of the algorithm creates lists of classes from an OWL file in RDF/XML syntax. Individual lists contain elements from a branch derived from each of the ontology top classes (a branch lists all subclasses of a top class; the number of branches is limited with the number of top classes multiplied with the number of subclasses). The lists are in ascending order so that the 0 index contains the leaf of the branch (the lowest subclass), and the higher indexes contain higher classes. In the second part of the algorithm, the tree of classes is filled with these lists. The algorithm performs the following steps:

- In step one it loads an OWL document and removes all non-essential information. Non-essential information includes user comments written in the XML file that do not syntactically match the ontology comments.
- In the second step, the algorithm acquires a list of all ontology classes. For each of them, the parent class is found and stores them in the form of "ClassName-ParentName" pairs. The first element is the class, the second is its parent class. Each pair is added to a list of pairs. A part of the list of pairs for the Pizza ontology is presented in Fig. 3a.
- In the third step, the algorithm loops over the list of pairs and performs data cleaning and verification procedures.
- In step four, the remaining pairs are joined in lists. The procedure starts with the first pair. It then scans the remaining pairs to find the pair where the parent class of the first pair is found as the class name. From that pair, the parent name can be extracted. Therefore, a first list can be formed with three members. The procedure is repeated until no more parents can be extracted. Then the entire procedure is repeated for all remaining pairs until all of them have been transformed to hierarchy lists. The merge procedure (pairs to lists) is presented in Fig. 3b. A part of the list of all merged lists is presented in Fig. 3c.
- In the final, fifth step the algorithm fill a tree by first creating the root of the tree called "Thing." ("Thing" being the superclass of all ontology classes.) The lists are then ordered according to length. The longest lists are ranked highest, therefore their elements are the first to be written to the tree. A part of the lists is shown in Fig. 3c. The number of elements in the longest list provides the depth of the tree. Each list is then processed individually. Each list is first checked for length. If the list is among the longest, the final element from it will become a child node of the root node ("Thing"), if that node has not been inserted already. The other elements from the list become its children (each a level lower). The procedure is run recursively for all the lists. The class hierarchy for the Pizza ontology is shown in Fig. 4.

C. *Class description*

Since the key to a successfully constructed DSL is a proper understanding of the source ontology, OntoP provides a description of all ontology classes. Individual classes are annotated with: equivalent classes, superclasses, members, disjoint classes, list of superclasses, comments and labels. Equivalent and superclasses descriptions are in Manchester OWL syntax. A description for the VegeterianPizza class, prepared by OntoP, is shown in Fig 4:

- VegeterianPizza is a type of Pizza with only Cheese or Vegetable toppings,

Figure 3. Building blocks: a part of the list of pairs for the Pizza ontology (a), the merge procedure (b) and a part of the list of all merged lists (c).

- VegeterianPizza has only one disjoint class; NonVegeterianPizza and
- superclasses of VegeterianPizza are Pizza and Thing.

IV. CONCLUSION AND FUTURE WORK

When working with OWL ontologies, a large set of tools exist that enable the creation, editing, reasoning over ontologies and the use of ontologies in various applications. In this paper, another parser for working with ontologies was presented: the OntoP ontology parser. OntoP was developed for the needs of a larger framework: Ontology2DSL.

Future work includes the separation of the OntoP parser from the Ontology2DSL framework so that it can be offered as a freely accessible tool for interested researchers.

ACKNOWLEDGMENT

This work is sponsored by bilateral project 'Language Patterns in Domain-specific Languages Evolution' between Slovenia (Grant No. BI-SK/11-12-011) and Slovakia (Grant No. SK-SI-0003-10).

REFERENCES

[1] L. Lacy, OWL: Representing Information Using the Web Ontology Language. Trafford Publishing, 2005.

[2] J. Hebeler, M. Fisher, R. Blace, and A. Perez-Lopez, Semantic Web Programming. Wiley Publishing, 2009.

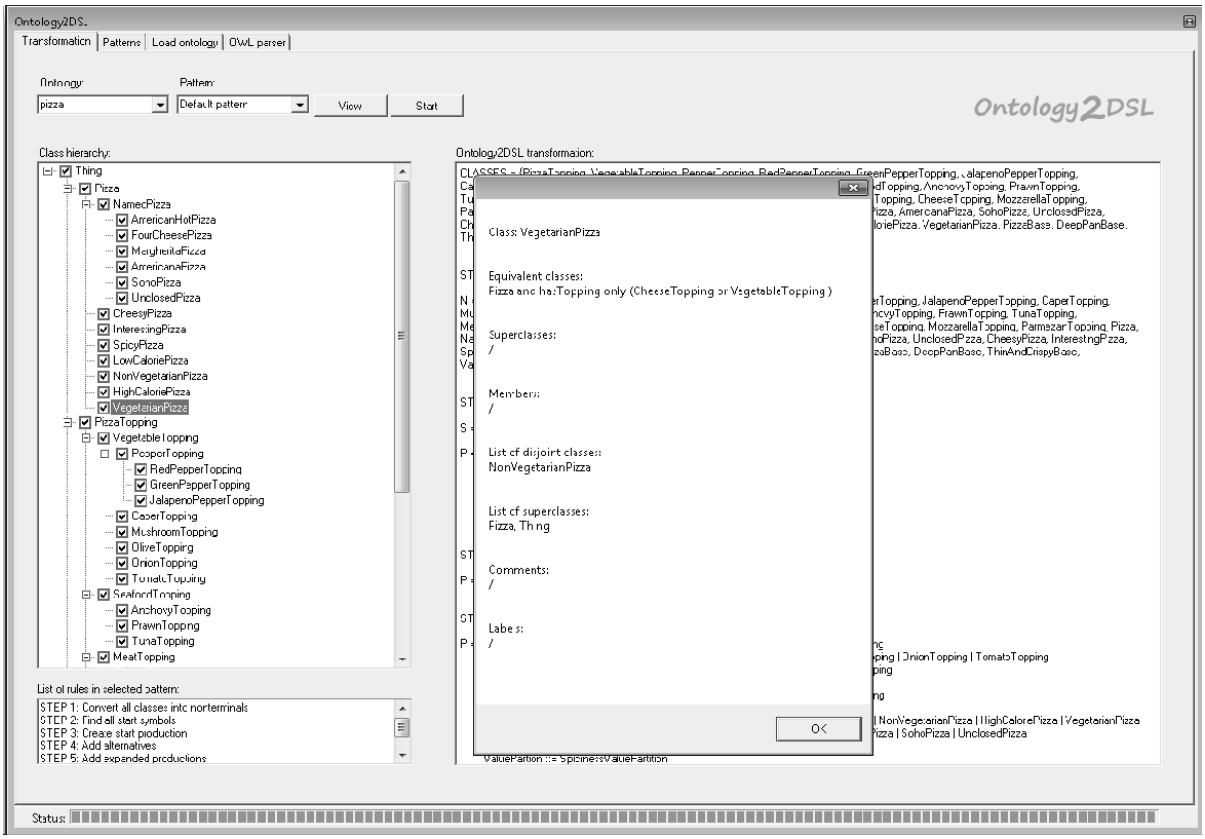


Figure 4. Ontology2DSL framework.

[3] R. Studer, R. Benjamins, and D. Fensel, Knowledge engineering: Principles and methods, Data & Knowledge engineering, vol. 25, pp. 161-197, April 1998.

[4] S. Staab and R. Studer, Handbook on Ontologies. Springer Verlag, 2009.

[5] Protege, http://protege.stanford.edu/.

[6] A. Kalyanpur, B. Parsia, and J. Hendler, A tool for working with web Ontologies, International Journal on Semantic Web and Information Systems, vol. 1, pp. 36-49, 2005.

[7] E. Sirin, B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz, Pellet: A practical OWL-DL reasoner, Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, pp. 51-53, June 2007.

[8] D. Tsarkov and I. Horrocks, FaCT++ Description Logic Reasoner: System Description, Proceedings of the International Joint Conference on Automated Reasoning, vol. 4130, pp. 292-297, August 2006.

[9] S. Bechhofer, R. Volz, and P. Lord, Cooking the Semantic Web with the OWL API, Proceedings of the First International Semantic Web Conference, pp. 659-675, October 2003.

[10] Jena, http://jena.sourceforge.net/.

[11] I. Čeh, M. Črepinšek, T. Kosar, and M. Mernik, Ontology Driven Development of Domain-Specific Languages, Computer Science and Information Systems, vol. 8, pp. 317-342, May 2011.

[12] A. V. Aho, M. S. Lam, R. Sethi, and J. D. Ullman, Compilers: Principles, Techniques, and Tools. Addison Wesley, 2007.

[13] M. Mernik, J. Heering, and A. M. Sloane, When and how to develop domain-specific languages, ACM Computing Surveys, vol 37, pp. 316-344, December 2005.

[14] G. Antoniou and F. van Harmelen, A Semantic Web Primer, second edition. The MIT Press, 2008.

[15] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, The description logic handbook: Theory, implementation and applications. Cambridge University Press, 2003.

[16] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuiness, P. F. Patel-Schneider, and L. A. Stein, Owl Web Language Reference, http://www.w3.org/TR/owl-ref/.

[17] P. F. Patel-Schneider, P. Hayes, and I. Horrocks, OWL Web ontology Language Semantics and Abstract Syntax, http://www.w3.org/TR/owl-semantics/.

[18] M. Horridge, N. Drummond, J. Hoodwin, A. Rector, R. Stevens, and H. Wang, The Manchester OWL syntax, Proceedings of the OWL Experiences and Directions Workshop, vol. 216, November 2006.

[19] M. K. Smith, C. W, and D. L. McGuinness, OWL Web ontology Language Guide, http://www.w3.org/TR/owl-guide/.

Advanced Safe Iterators for the C++ Standard Template Library

Norbert Pataki
Department of Programming Languages and Compilers,
Eötvös Loránd University
Pázmány Péter sétány 1/C H-1117 Budapest, Hungary
Email: patakino@elte.hu

***Abstract*—The C++ Standard Template Library is the flagship example for libraries based on the generic programming paradigm. The usage of this library is intended to minimize classical C/C++ error, but does not warrant bug-free programs. Furthermore, many new kinds of errors may arise from the inaccurate use of the generic programming paradigm, like dereferencing invalid iterators or misunderstanding remove-like algorithms.**

In this paper we present some typical scenarios, what are mean risk from the view of program safety. We present an approach that can be used for developing safe iterators to avoid runtime errors. These iterators are able to manipulate the container directly, hence they cannot result in undefined behaviour when an algorithm needs to add elements to the container or delete elements from the container.

I. Introduction

The *C++ Standard Template Library* (STL) was developed by *generic programming* approach [1]. In this way containers are defined as class templates and many algorithms can be implemented as function templates. Furthermore, algorithms are implemented in a container-independent way, so one can use them with different containers [2]. C++ STL is widely-used because it is a very handy, standard library that contains beneficial containers (like list, vector, map, etc.), a lot of algorithms (like sort, find, count, etc.) among other utilities [3].

The STL was designed to be extensible. We can add new containers that can work together with the existing algorithms. On the other hand, we can extend the set of algorithms with a new one that can work together with the existing containers. Iterators bridge the gap between containers and algorithms [4]. The expression problem is solved with this approach [5]. STL also includes adaptor types which transform standard elements of the library for a different functionality [6].

However, the usage of C++ STL does not guarantee bugfree or error-free code [7]. Contrarily, incorrect application of the library may introduce new types of problems [8].

One of the problems is that the error diagnostics are usually complex, and very hard to figure out the root cause of a program error [9], [10]. Violating requirement of special preconditions (e.g. sorted ranges) is not checked, but results in runtime bugs [11]. A different kind of stickler is that if we have an iterator object that pointed to an element in a container, but the element is erased or the container's memory allocation has been changed, then the iterator becomes *invalid*. Further reference of invalid iterators causes undefined behaviour [12].

Another common mistake is related to removing algorithms. The algorithms are container-independent, hence they do not know how to erase elements from a container, just relocate them to a specific part of the container, and we need to invoke a specific erase member function to remove the elements phisically. Therefore, for example the `remove` and `unique` algorithms do not actually remove any element from a container [13].

The previously mentioned `unique` has uncommon precondition. Equal elements should be in consecutive groups. In general case, using `sort` algorithm is advised called before the invokation of `unique`. However, `unique` cannot result in an undefined behaviour, but its result may be counter-intuitive at first time.

Some of the properties are checked at compilation time. For example, the code does not compile if one uses sort algorithm with the standard list container, because the list's iterators do not offer random accessibility [14]. Other properties are checked at runtime. For example, the standard vector container offers an `at` method which tests if the index is valid and it raises an exception otherwise [15].

Unfortunately, there is still a large number of properties are tested neither at compilation-time nor at run-time. Observance of these properties is in the charge of the programmers. On the other hand, type systems can provide a high degree of safety at low operational costs. As part of the compiler, they discover many semantic errors very efficiently.

Associative containers (e.g. `multiset`) use functors exclusively to keep their elements sorted. Algorithms for sorting (e.g. `stable_sort`) and searching in ordered ranges (e.g. `lower_bound`) typically used with functors because of efficiency. These containers and algorithms need *strict weak ordering*. Containers become inconsistent, if used functors do not meet the requirement of strict weak ordering [16].

Certain containers have member functions with the same names as STL algorithms. This phenomenon has many different reasons, for instance, efficiency, safety, or avoidance of compilation errors. For example, as mentioned, list's iterators cannot be passed to `sort` algorithm, hence code cannot be compiled. To overcome this problem list has a member function called `sort`. List also provides `unique` method. In these cases, although the code compiles, the calls of member

functions are preferred to the usage of generic algorithms.

In this paper we argue for some extensions for the C++ STL to make it safer. Our extensions help to avoid runtime problems: one can use copying algorithms with our copy-safe iterators and the misuse of remove-like algorithms can be evaded.

This paper is organized as follows. In section II we present some motivating examples. In section III we present the implementation of erasable iterators. In section IV we present a new kind of iterator that help to overcome some copy-related problems. Finally, this paper concludes in section V.

II. Motivation

STL's `copy` and `transform` algorithm can be used to copy an input range of objects into a target range. These algorithms neither allocate memory space nor call any specific inserter method while coping elements. They assume that the target has enough, properly allocated elements where they can copy elements with `operator=`. Inserter iterators can enforce to use `push_back`, `push_front` or `insert` method of containers. But these algorithms cannot copy elements into an empty list, for instance. They do not know how to insert elements into the empty container. The following code snippet can be compiled, but it results in an undefined behaviour [17]:

```
std::list<int> li;
std::vector<int> vi;
v.push_back( 3 );

std::copy( vi.begin(),
           vi.end(),
           li.begin() );
```

However, there are some adaptors in the library to overcome this situation: `back_inserter` and `front_inserter` adaptors. On the other hand, they cannot change the elements of a container, only add new element to the container [18]. However, we have developed a technique that is able to emit compilation warnings if the usage of the `copy` algorithm may be erroneous [19].

Another common mistake is related to removing algorithms. The algorithms are container-independent, hence they do not know how to erase elements from a container, just relocate them to a specific part of the container, and we need to invoke a specific erase member function to remove the elements phisically. Therefore, for example the `remove` and `unique` algorithms do not actually remove any element from a container [13]. Let us consider the following code snippet:

```
std::vector<int> v;
for( int = 1; i <= 10; ++i )
  v[i] = i;

v[3] = v[5] = v[7] = 99;

std::remove( v.begin(), v.end(), 99 );

std::cout << v.size();
```

In contrast to the name of the algorithm, the container's size is unchanged. The remaining elements has been moved to front of the container, but the tail is also unchanged. The result of this algorithm may be counter-intuitive at first time. The proper usage of the remove is called *erase-remove idiom*:

```
v.erase( std::remove( v.begin(),
                      v.end(),
                      99 ),
         v.end() );
```

III. Erasable iterators

In this section we present our approach to develop iterators that are able manipulate the container and remove elements from it [20].

First, we add a new inner class to the `vector` container. This class is called `erasable_iterator`: this is quite similar to the standard `iterator` class, but it has a *pointer* to the container and a new member function called `erase`. This method accesses the member functions of the container via the pointer. Only point is that the method has to avoid invalidation of the iterator. The container's member function `ebegin` returns an erasable iterator to the first element, and its method `eend` returns an erasable iterator to the end of the sequence, respectively.

```
typedef
  std::random_access_iterator_tag
  ran_acc_tag;

template <class T,
          class Alloc = std::alloc<T> >
class vector
{
  T* p;
  int s, cap;
  // usual vector's members, typedefs,
  // classes, operators

public:

  class iterator:
    public
     std::iterator<ran_acc_tag,
                   T>
  {
  protected:
     T* p;
     // usual operators...
  };

  class erasable_iterator:
    public iterator
  {
    vector<T>* v;
  public:
    erasable_iterator( T* p,
                       vector<T>* vt ):
```

```
        iterator( p ), v( vt )
    { }

    void erase()
    {
      T* tmp = iterator::p + 1;
      v->erase( *this );
      iterator::p = tmp;
    }
  };

  erasable_iterator ebegin()
  {
    return erasable_iterator( p, this );
  }

  erasable_iterator eend()
  {
    return erasable_iterator( p + s,
                              this );
  }

};
```

This technique can be transformed to other containers, too.

We have to rewrite the remove-like algorithms to take advantage of the erasable iterators. For example, we can write the following algorithm:

```
template <class EraseIter, class T>
void remove( EraseIter first,
             EraseIter last,
             const T& t )
{
  while( first != last )
  {
    if ( t == *first )
    {
      first.erase();
    }
    else
    {
      ++first;
    }
  }
}
```

IV. Copy-safe iterators

In this section we present our implementation of copy-safe iterators.

This iterator type is also similar to `iterator` type of the container. This kind of iterator also has a *pointer* to the container. When a safe pointer is dereferenced (ie. its `operator*` is called, it can invoke the container's `push_back` method and add new element to the `vector` if necessary. Hence, if this kind of iterators is in use it causes no runtime problems if someone copies elements into an empty `vector`. Our implementation is able to detect if the client uses problematic iterators for copying ranges [19]. The container's member function `cbegin` returns a copy-safe iterator to the first element, and its method `cend` returns a copy-safe iterator to the end of the sequence, respectively.

```
template <class T,
          class Alloc = std::alloc<T> >
class vector
{
  // usual vector's members, typedefs,
  // classes like above

  class copy_iterator: public iterator
  {
    vector<T>* v;
  public:
    copy_iterator( T* ptr,
                   vector<T>* vt )
    : iterator( ptr ), v( vt )
    { }

    T& operator*()
    {
      if ( *this == v->end() )
      {
        v->push_back( T() );
        iterator::p = &( v->back() );
      }
      return iterator::operator*();
    }
  };

  copy_iterator cbegin()
  {
    return copy_iterator( p, this );
  }

  copy_iterator cend()
  {
    return copy_iterator( p + s, this );
  }

};
```

This technique can be transformed to other containers, too.

Modification of any algorithms is not necessary because these iterators can be work with standard copying algorithms, such as `copy` or `transform`. For instance, the following code snippet shows the usage that cannot be implemented in the original STL way:

```
std::vector<int> vi;
// ...
std::list<int> li;
// ...
std::copy( li.begin(),
           li.end(),
           vi.cbegin() );
```

This invokation of `copy` algorithm overwrites all the exist-

ing elements in the `vector`, and added more new elements to the vector, if necessary.

Limitations can be mentioned with this approach. However, `vector` does not offer `push_front` method, the `copy_iterator` should be parametrized with strategy of adding new element to container. *Function objects* (also known as *functors*) make the library much more flexible without significant runtime overhead. They parametrize user-defined algorithms in the library, for example, they determine the comparison in the ordered containers or define a predicate to find.

The iterator always executes a check when it is dereferenced, it has runtime overhead. However, it guarantees safety, and original non-copier iterators are available, too. The runtime overhead should be measured [12].

V. Conclusion

STL is the most widely-used library based on the generic programming paradigm. STL increases efficacy of C++ programmers mightily because it consists of expedient containers and algorithms. It is efficient and convenient, but the incorrect usage of the library results in weird or undefined behaviour [15].

In this paper we present some examples that can be compiled, but at runtime their usage is defective. We argue for some new extensions of the containers. New kind of iterators are able to overcome problematic situations. The limitations of these iterators are also discussed.

Acknowledgment

The Project is supported by the European Union and co-financed by the European Social Fund (grant agreement no. TÁMOP 4.2.1./B-09/1/KMR-2010-0003).

References

[1] B. Stroustrup, *The C++ Programming Language*, special ed. Reading, MA: Addison-Wesley, 2000.

[2] D. R. Musser and A. A. Stepanov, "Generic programming," in *Proc. of the International Symposium ISSAC'88 on Symbolic and Algebraic Computation*, ser. Lecture Notes in Comput. Sci., vol. 358, 1988, pp. 13–25.

[3] M. H. Austern, *Generic Programming and the STL: Using and Extending the C++ Standard Template Library*. Reading, MA: Addison-Wesley, 1998.

[4] G. Dévai and N. Pataki, "Towards verified usage of the C++ Standard Template Library," in *Proc. of The 10th Symposium on Programming Languages and Software Tools (SPLST) 2007*, 2007, pp. 360–371.

[5] M. Torgersen, "The expression problem revisited – four new solutions using generics," in *Proc. of European Conference on Object-Oriented Programming (ECOOP) 2004*, ser. Lecture Notes in Comput. Sci., vol. 3086, 2004, pp. 123–143.

[6] A. Alexandrescu, *Modern C++ Design*. Reading, MA: Addison-Wesley, 2001.

[7] G. Dévai and N. Pataki, "A tool for formally specifying the C++ Standard Template Library," *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae, Sectio Computatorica*, no. 31, pp. 147–166, 2009.

[8] P. Pirkelbauer, S. Parent, M. Marcus, and B. Stroustrup, "Runtime concepts for the C++ Standard Template Library," in *Proc. of the 2008 ACM Symposium on Applied Computing*, 2008, pp. 171–177.

[9] L. Zolman, "An stl message decryptor for visual c++," *C/C++ Users Journal*, vol. 19(7), pp. 24–30, 2001.

[10] I. Zólyomi and Z. Porkoláb, "Towards a general template introspection library," in *Proc. of Generative Programming and Component Engineering: Third International Conference (GPCE 2004)*, ser. Lecture Notes in Comput. Sci., vol. 3286, 2004, pp. 266–282.

[11] D. Gregor, J. Järvi, J. Siek, B. Stroustrup, G. D. Reis, and A. Lumsdaine, "Concepts: linguistic support for generic programming in C++," in *Proc. of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications (OOPSLA 2006)*, 2006, pp. 291–310.

[12] N. Pataki, Z. Szűgyi, and G. Dévai, "Measuring the overhead of C++ Standard Template Library safe variants," *Electronic Notes in Theoretical Computer Science*, no. 264(5), pp. 71–83, 2011.

[13] S. Meyers, *Effective STL – 50 Specific Ways to Improve Your Use of the Standard Template Library*. Reading, MA: Addison-Wesley, 2001.

[14] J. Järvi, D. Gregor, J. Willcock, A. Lumsdaine, and J. Siek, "Algorithm specialization in generic programming: challenges of constrained generics in C++," in *Proc. of the 2006 ACM SIGPLAN conference on Programming language design and implementation (PLDI 2006)*, 2006, pp. 272–282.

[15] N. Pataki, Z. Porkoláb, and Z. Istenes, "Towards soundness examination of the C++ Standard Template Library," in *Proc. of Electronic Computers and Informatics (ECI 2006)*, 2006, pp. 186–191.

[16] N. Pataki, "Advanced functor framework for C++ Standard Template Library," *Studia Universitatis Babeş-Bolyai, Informatica*, no. LVI(1), pp. 99–113, 2011.

[17] N. Pataki, Z. Szűgyi, and G. Dévai, "C++ Standard Template Library in a safer way," in *Proc. of Workshop on Generative Technologies 2010 (WGT 2010)*, 2010, pp. 46–55.

[18] N. Pataki, "C++ Standard Template Library by ranges," in *Proc. of 8th International Conference on Applied Informatics (ICAI 2010)*, vol. 2, pp. 367–374.

[19] N. Pataki and Z. Porkoláb, "Extension of iterator traits in the C++ Standard Template Library," in *Proc. of the Federated Conference on Computer Science and Information Systems*, 2011, pp. 919–922.

[20] T. Becker, "STL & generic programming: writing your own iterators," *C/C++ Users Journal*, vol. 19(8), pp. 51–57, 2001.

Analysing Erlang BEAM files

Mátyás Karácsonyi and Melinda Tóth
Eötvös Loránd University
Faculty of Informatics
Department of Programming Languages and Compilers
Email: {k_matyas,toth_m}@inf.elte.hu

***Abstract*—Different static analyser tools have been developed to support program transformations or system comprehension. RefactorErl is a source code analyser and transformer tool for Erlang that provides both facilities. RefactorErl stores, manipulates and analyses the source code in a Semantic Program Graph that is based on an abstract syntax tree (AST). However if the source code is not available, the tool can not produce the syntax tree of the program. The accuracy of the different source code analysis used by RefactorErl depends on the represented information about the software, thus we provide a solution for static analysis of Erlang modules even when the source code of the module is not available: using the compiled Erlang BEAM files we can build the AST and the Semantic Program Graph.**

I. Introduction

Several static source code analyser tools have been developed to support software comprehension, program transformation, testing, bug detection etc. Those tools mainly concentrate on languages that are heavily used in the industry.

Our research focuses on the programming language Erlang [1], that has been getting widespread in the last decade. Erlang is a dynamically typed functional programming language designed to develop highly concurrent, fault tolerant, distributed systems, with soft-real time characteristic. Huge commercial and open source applications have been developed using Erlang with millions of lines of code, therefore a tool that can help the developers in everyday programming task is very useful. We have been developing an open source static source code analyser and transformer tool for this purpose, called RefactorErl [2].

Like most of the static analyser tools, RefactorErl builds and manipulates an intermediate representation of the source code. The tool represents the source code in a Semantic Program Graph (SPG) [3] that has three layers: a lexical layer containing all of the token, whitespace and comment information, the syntactic layer containing the abstract syntax tree of the program and a semantic layer. The semantic analyser framework of RefactorErl evaluates different static analysis on the SPG and adds new nodes and links to the graph, representing data-flow, variable binding information, and so on. The result and the accuracy of the performed static semantic analysis are highly dependent on the available source code. For instance, when a third party application is used under development and the source code of this application is not available, RefactorErl can not determine how the data flows through a function used from the provided interface of the third party application.

Supported by KMOP-1.1.2-08/1-2008-0002, European Regional Development Fund (ERDF) and Ericsson Hungary

Our goal is to develop an extension to RefactorErl, which is capable to analyse the software when the source code is not available. Therefore we want to develop a tool to analyse `.beam` files of the Erlang modules and build the AST based on this analysis. The `.beam` file of a module must be available for the developers even in case of a used third party application.

BEAM is the file format of the compiled Erlang modules and also the name of the Erlang Virtual Machine [4], [5]. The control language of the Erlang VM is the BEAM assembly. Since the Erlang compiler performs some optimisations when creating the BEAM assembly code of an Erlang module, building the original syntax tree is not possible, but the resulted tree is semantically and functionally equivalent with the original one.

We have studied the `.beam` files and developed two methods to build the abstract syntax tree (AST) of the Erlang modules from the `.beam` files. There are two different ways of compiling the Erlang modules: with or without debug information. In the former case the compiler adds the parse tree of the module to the `.beam` file. The first method describes the AST generation from the parse tree of the compiler and the other method describes the AST generation from the BEAM assembly code. Based on the AST we can build the SPG and the further performed static semantic analysis became more precise.

The next sections present the BEAM analysis and the AST construction when the debug information is available in the compiled BEAM files (Section II) and also when it is compiled without debug information (Section III); Section IV contains the related work; and Section V concludes the paper.

In the next sections we will present our analysis and the different representations of the source code based on the factorial from the Erlang module introduces in Figure 1. This module defines its name `math`, exports the function `fact/1`, defines the `END` macro and the `fact` function. The `fact` function takes an argument: `N`, if the value of the argument is `0` or `1` the function returns `1`, otherwise it evaluates the second function clause: binds the result of the case expression to the variable `NewN` and then multiply it with `N`. If the value of the argument matches to the value of the `END` macro the case expression returns `1`, otherwise it calls the `fact` function with `N-1` and returns the result of this function application.

The syntax tree of this `math` module is shown in Figure 2.

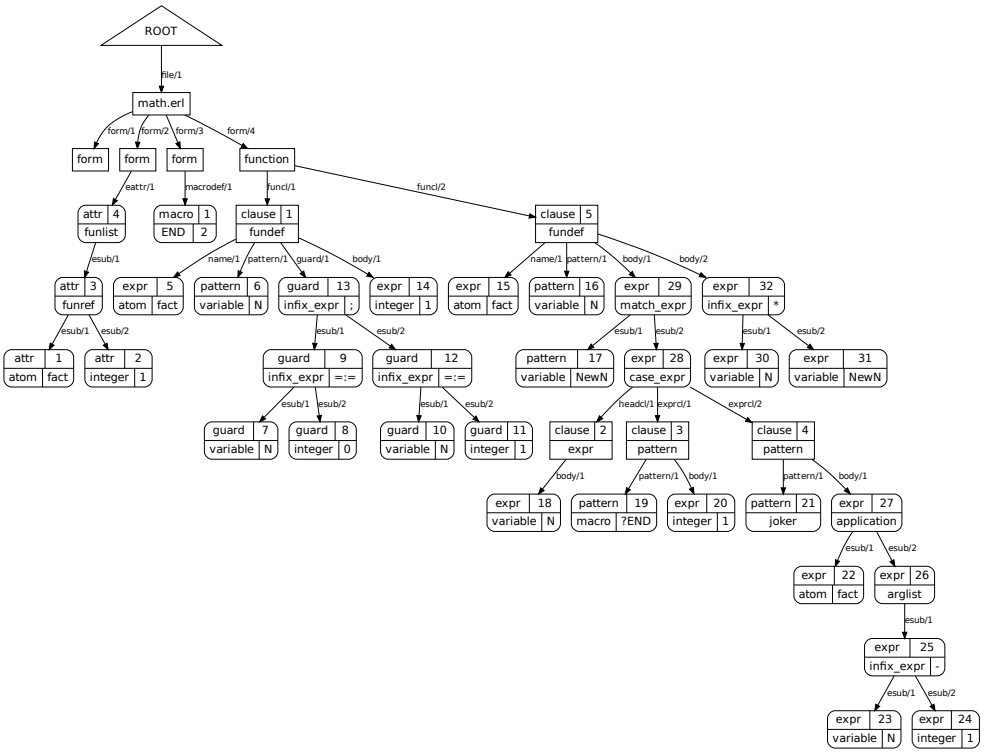


Fig. 2. The syntax graph of the `fact` function

```
-module(math).

-export([fact/1]).

-define(END, 2).

fact(N) when N =:= 0;
            N =:= 1->
  1;
fact(N) ->
  NewN =
    case N of
      ?END -> 1;
      _    -> fact(N - 1)
    end,
  N * NewN.
```

Fig. 1. The Erlang code of the `math` module

II. Analysing the Abstract Code

During compilation the standard Erlang compiler generates the parse tree (abstract code) of the input module [6], which can be included into the `abstract_code` chunk [7] of the BEAM file by the `debug_info` compiler option. If the abstract code of a module is stored into its BEAM file, the parse tree can be read out. As the abstract code is built-up after the lexical and syntactical analysis, the preprocessor constructions (macro applications and the file includes) are evaluated before the abstract code of the module is generated. In addition there is no information about the original code indentation in abstract code.

A. Abstract Code

The Abstract Code is represented as a list of tuples (an ordered list of elements), where every element represent a syntactical structure. The first value of each element defines its type, the second is the line number of the original language construction in the source file. The rest of the tuple is specific for every language element. The first element of an Abstract Code sequence determines the file name, the last is the `{eof, Line}` tuple, which closes the module.

B. Transformation

The recovered code sequence is transformed to the syntax tree description language of the RefactorErl [8], which can be loaded into the semantic graph of RefactorErl. The transformer function generates a syntax tree definition from every abstract code expression. For simple language constructs (like simple data types, variables etc.) there are minor differences in the two representation, therefore the conversion is straightforward.

But in most of the cases complex transformation is necessary, for example the abstract code does not contain any type information about the module attributes.

The most difficult part is to transform those expressions, where the two representations differ. The abstract code groups the language elements by their usage, whereas the RefactorErl concentrates on their syntactic structure. For example, if the function call is qualified, the Abstract Code has the type `remote` and stores the module and function identifier, but in RefactorErl the module name and the function name are arguments of the `:` infix operator. In addition, the representation of the list expressions also differs. The Abstract Code represents the list as a recursive structure, whereas the RefactorErl unfold them to head and tail, where the head elements are flattened out.

From the generated syntax tree definition the abstract code analyser imports it into the Semantic Program Graph, and the semantic analyser framework of the RefactorErl performs several static analysis, including the pretty printer, which automatically formats the source code. We note here, that RefactorErl does not generate source files after the importing, but it is possible to generate the source code from the Semantic Program Graph and write it to a file.

Few steps of recovering the syntax tree of the `fact` function using the abstract code is presented on Figure 3., where similarities and differences of the two representation can be seen. The resulted graph (Figure 4.) is similar to the original (Figure 2.) thanks to the available information in the Abstract Code. The only differences are that the `?END` macro is substituted with the value `2` and no macro definition is generated.

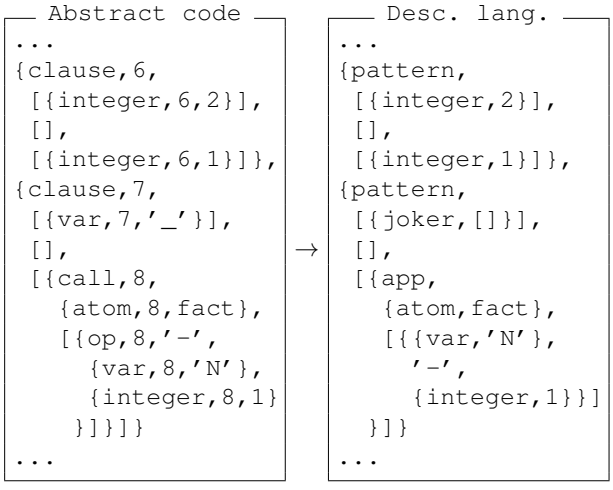


Fig. 3. Transforming Abstract Code to RefactorErl syntax tree description language (the clauses of the `case` expression)

III. Analysing the BEAM Files

The BEAM assembly is an imperative operation control language, which is used by the BEAM virtual machine [4], [5]. In most cases Erlang developers do not include the abstract code of their module to the BEAM files, therefore, the only way to recover semantic information from them is to analyse the BEAM assembly code of the module. It was a challenge to survey the BEAM assembly language, because its specification is not available for publicity, therefore we had to examine its structure first [8]. In addition we had to find a way to generate functional language source code from an imperative operation control language.

The outcome of our research is an Erlang decompiler. The analyser first constructs the Control-Flow Graph (CFG) and the Data-Flow Graph (DFG) of the functions from the analysed BEAM file. The type information is directly available in the BEAM assembly structure and this information is stored in the DFG for further calculation.

A. BEAM assembly

The BEAM assembly is an imperative programming language with restricted language constructs. For example the BEAM assembly does not store the record definitions and usages, because the records are represented with tuples in Erlang, where the first element of the tuple is the record name and the following elements are the values of the record fields. In BEAM assembly every module is composed from five parts, where each parameters are lists of tuples (except the module name):

- module name,
- exported functions,
- optional module attributes ("wild attributes"),
- compiler information,
- code table (function definitions).

To analyse a BEAM assembly code sequence we had to group its commands. However the operation control language supports every Erlang construct, but some of them are grouped for optimisation reasons, thus make the analysis harder.

a) Types: In BEAM assembly we divide the data elements to constants and dynamic elements and their types to simple (atom, number) and complex (binary, list, tuple). We call a memory element dynamic, if its type is complex and some parts of it come from parameters or they are the return value of a program block. The constant complex typed elements do not have any keyword to differentiate them. Each complex element has a type constructor and selector.

b) Function calls: The BEAM assembly defines static (`call` and `call_ext`) and dynamic (`apply`) function call commands, as well as in Erlang. If a function has parameters they have to be copied to the `x` register. In case of a dynamic function call the module and the function name also have to be copied to the register. The return value of a function call is in the zeroth index of the `x` register.

c) Conditionals: The BEAM assembly groups the conditionals for optimisation reasons. It has only pattern matching and selection operations which describe the `case` expressions, the `if` expressions and the clauses of the function definitions. If a conditional return value is false, the operation jumps to a label where the program can execute the next operation.

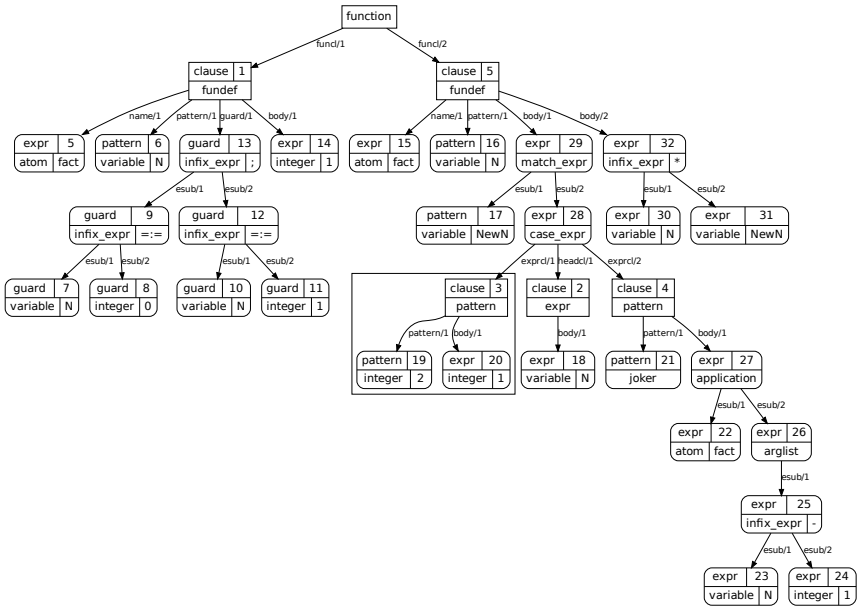


Fig. 4. The recovered syntax graph of the `fact` function using the Abstract Code

d) Others: There are several language elements which need special analysis. For example the binary/list generator implementations are transformed to local functions which get called from the defining function. In addition the `try` and `receive` statements also have different structure.

B. Building the AST

Some information can be recovered from an Erlang BEAM file, without complex analysis e.g, exported functions, wild attributes etc., but building a semantically equivalent function from a BEAM assembly function block requires a more sophisticated analysis using decompilation techniques. Therefore we have to build the Control- and Data-Flow Graphs of the function blocks and generate their syntax trees by traversing these graphs.

e) Control-Flow Graph (CFG): The CFG is a directed labelled graph (Figure 5.), which represents every possible execution plan of the analysed function. The CFG stores the most important commands of a code sequence as nodes (it neglects e.g: special virtual machine control operations etc.) and the successive commands are linked with edges. Each node of the built CFG has one of these properties, depending on the stored command:

- **label:** determines a logical block, where the control can be passed. The root of the CFG of every function block is a label, which stores its starting label.
- **conditional:** is a forkable node. Only the `test` or `select_val` commands can have this property. The edges of the conditionals point to a label node where the operation can be continued.
- **type selector:** are logically part of the conditionals, but they cannot fork.
- **sequence:** contains every command which is not part of the former ones e.g, type constructors, function calls etc.

When the CFG builder algorithm reaches the BEAM assembly representation of the `case` expression in the `fact` function (Figure 1.) a conditional typed node is created, which has a `true` and a `false` branch (Figure 5.). In the `true` subgraph a constant integer get copied to the `{x,0}` memory element (`move, {integer, 1}`) then jumps to label 4. The `false` branch describes the mathematical operation `N-1` in step 1 (represented by the node `'-' [{x,0},{integer,1}],{x,0}`) and a function call in step 2 (`call fact/1`). Both subgraphs pass the control to the same logical block (denoted with `label,4`).

f) Data-Flow Graph (DFG): The DFG is an directed labelled graph (Figure 6.) which stores the content and type information of the memory elements. It can be visualised as a directed graph and each edge of the graph point from a memory element to a syntax sub-tree definition. This graph is constructed in parallel with the CFG hence the BEAM assembly commands can clarify the content and the type information of a memory element. So no type analysis is required for the decompiler, because its functionality is replaced by the data-flow analyser. If no type information can be recovered about a memory element the syntax tree generator will consider it as a variable.

Initially the DFG of the analysed function contains only the formal parameters of the function without any type information. During the building of the CFG this graph get extended

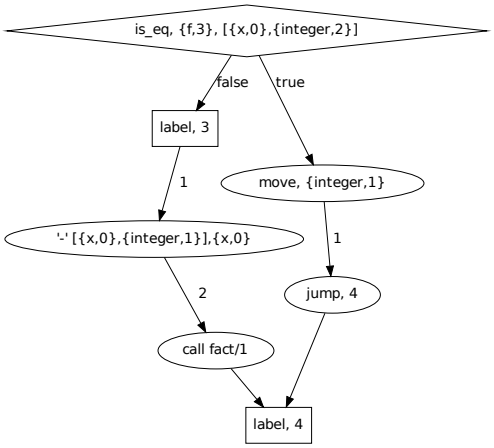


Fig. 5. Sub-control-flow graph of the `fact` function

with type information or newer memory constructions. If the CFG get forked during its building-up the DFG of the analysed function get cloned for every new subgraph, so the latter memory operations do not get mixed up. In addition, if the DFG builder algorithm notices a function call, it updates the `{x,0}` memory element node of the graph where the return value of the called function will be stored (without any type information).

In our example when the CFG forks (Figure 5.) the DFG of the analysed function (Figure 6.), which constains only one memory element without any type information, get cloned to a `true` and a `false` named graphs. In both cases the value of the function argument copied to the `x` register at first. In the `true` graph a constant integer get copied to the `{x,0}` memory element in step 1. The `false` graph describes that in step 1 the original value of the memory element is substracted by 1 then in step 2 the `fact` function get called, so the value of the `{x,0}` memory element has to be replaced by a no typed element (labeled with `ret_val`, that represents the return value of the function call).

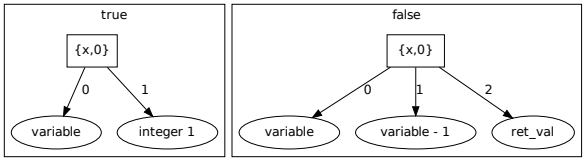


Fig. 6. Data-Flow sub-Graphs of the factorial `fact` function

C. Syntax tree generation

The syntax tree generator algorithm recovers the module parameters (e.g: exported functions etc.) at first, then starts to analyse the generated CFGs and DFGs. Eventually it traverses the CFG of each function from its root and generates syntax tree definitions from the results.

The following rules are used by the analysis and the generation:

- if the CFG gives the execution to another function (e.g: function call, message sending etc.), the values of the actual parameters are imported from the DFG;
- if the CFG node is a `receive` or `try` structure the frame and the possible return values are generated. The generation of these structures should be imagined as the generation of a function block;
- if the CFG node is a forkable node then it defines a conditional expression. In this case a coherent subgraph have to be cut from the CFG which has to contain nodes with conditional or type selector property. The result will be a tree which can determine a `case`, an `if` or a function header expression (the header and the guard of a function). To select the corresponding language element we have to analyse the context in the graph:
  - if the conditional does not have any previous command it is a function header;
  - if the conditional examines only one memory element (or its children in the DFG) it is a `case` expression;
  - every other conditional have be considered as an `if` expression.
- the syntax tree generator algorithm counts the references of a memory element in order to decide whether a new variable should be introduced or not.

Building a binary/list generator is not straightforward, because they are represented as separated local functions in the BEAM assembly. Until we find out a general process to identify these code parts we represent them as function calls (with the corresponding parameters) in the generated AST.

Finally the resulted syntax tree definition is imported to the Semantic Program Graph of RefactorErl and the asynchronous incremental semantic analyser framework build the whole SPG from it.

Having holded the above rules the resulted syntax tree of the recovered function is presented on Figure 7. which differs from the original graph (Figure 2.) in several points. For instance, the name of the used variables in the case expression can not be restored from the BEAM file, therefore the name of the variables are generated (`Var1, Var2`). Another difference that we can not distinguish the guard `when N=:=0 orelse N=:=1 -> ...` from the function clause headers (`fact(0) -> ...; fact(1) -> ...`) and the latter one generated in this example.

IV. Related Work

The theory of decompiling is a widely researched topic for imperative programming languages [9], [10], and some decompilers already developed [11], [12] for them. The more structure information is available in the operation control

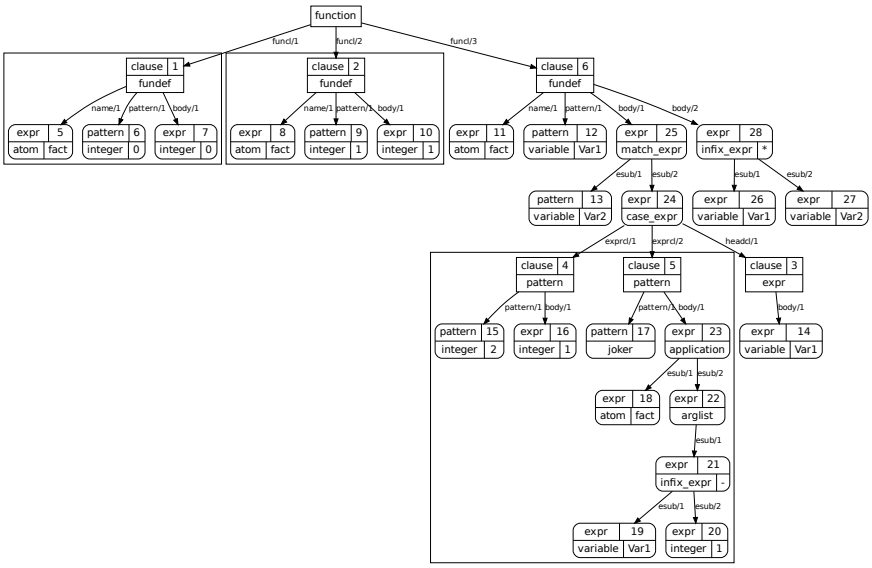


Fig. 7. The recovered syntax graph of the `fact` function using the BEAM assembly

language of a platform, the more resembling is the decompiled source code to the original. For example the Intel x86 assembly language does not store any type information about a memory segment, so a C/C++ decompiler has to have a complex type analyser which can calculate the used types by memory operations. Therefore it might not be accurate for complex applications. On the other hand in Java the bytecode of the virtual machine unambiguously determines the type of the used variables, accordingly a Java decompiler can recover nearly the original source code (except the optimised parts e.g: unused variables etc.).

About ten years ago some programmers came out with the idea of decompiling Erlang programs, but the documentation of the BEAM virtual machine was not accomplished, therefore the project was cancelled. Since then the official BEAM assembly specification has not been published.

V. Conclusions

In this paper we presented a special decompiling method for Erlang. Instead of generating the source code of Erlang module from its compiled BEAM assembly file, we build the Abstract Syntax Tree of the compiled program and load it to the Semantic Program Graph of RefactorErl.

The main motivation of our work was to make the result of the analysis of RefactorErl precise as much as possible. The accuracy of the analysis depends on the range of loaded modules, so adding the used third-party applications to the graph – even when the source code is not available – is important.

We studied two different ways to build the syntax tree. If the module is compiled with debug information, we build the syntax tree from the abstract code of the module available in the BEAM file. Otherwise we have to analyse the BEAM assembly file, build its Control and Data Flow Graph, and create the syntax tree from it. The main difficulty in the latter case is to build the syntax tree of a functional program from an imperative style assembly code.

The presented method is capable to apply it on industrial scale software.

References

[1] "Open Source Erlang," 2010, http://www.erlang.org/.
[2] "RefactorErl Home Page," 2010, http://plc.inf.elte.hu/erlang/.
[3] Z. Horváth, L. Lövei, T. Kozsik, R. Kitlei, A. N. Víg, T. Nagy, M. Tóth, and R. Király, "Modeling semantic knowledge in Erlang for refactoring," in *Knowledge Engineering: Principles and Techniques, Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques, KEPT 2009*, ser. Studia Universitatis Babe-Bolyai, Series Informatica, vol. 54(2009) Sp. Issue, Cluj-Napoca, Romania, Jul 2009, pp. 7–16.
[4] J. Armstrong, "A history of Erlang," in *Proceedings of the third ACM SIGPLAN conference on History of programming languages*, ser. HOPL III. New York, NY, USA: ACM, 2007. [Online]. Available: http://doi.acm.org/10.1145/1238844.1238850
[5] J. Armstrong and R. Virding, "The evolution of the Erlang VM," 2010, talk at Erlang Factory, London.
[6] *The Abstract Format*, http://www.erlang.org/doc/apps/erts/absform.html.
[7] B. Gustavsson, *File format for Beam R5 and later*, 2000, http://www.erlang.se/ bjorn/beam_file_format.html.
[8] M. Karácsonyi, "BEAM fájlok elemzése," National Scientific Students' Associations Conference, ELTE, Budapest, Hungary (second award), April, 2011.
[9] C. Cifuentes and K. J. Gough, "Decompilation of binary programs," *Softw. Pract. Exper.*, vol. 25, pp. 811–829, July 1995. [Online]. Available: http://portal.acm.org/citation.cfm?id=213593.213604
[10] G. Nolan, *Decompiling Java*. APress, 2004.
[11] "Java decompiler," 2010, http://java.decompiler.free.fr/.
[12] C. Cifuentes, "The dcc Decompiler," 2002, http://archive.itee.uq.edu.au/ cristina/dcc.html/.

Generating member functions and operators by tagged fields in a C++

Zalán Szűgyi
Department of Programming Languages and Compilers,
Eötvös Loránd University
Pázmány Péter sétány 1/C H-1117 Budapest, Hungary
Email: lupin@ludens.elte.hu

Gergely Klár
Department of Algorithm and Applications,
Eötvös Loránd University
Pázmány Péter sétány 1/C H-1117 Budapest, Hungary
Email: tremere@inf.elte.hu

***Abstract*—C++ is still one of the most recently used programming language in software development. Its standard library, the STL, provides a variety of useful containers and algorithms. As the STL is implemented in generic programming paradigm it is easy to apply user defined types on its containers and algorithms. However there are some restriction of a types used with STL: the elements stored in a set must be less then comparable, elements searched by algorithm find must be equality comparable etc. In a general way, the programmers can define own operators for their types, or they can customize the STL with proper functor types, however, this solution often laborious.**

We propose a tagging mechanism that generates the necessary operators. In addition not only operators but other member functions can be generated, such as: dumping variables, getters, setters, etc. Our solution based on standard of C++, thus no external tool is needed to compile to source code.

I. Introduction

The *Standard Template Library* (STL) [1], [2] of the C++ programming language [3] contains a variety of beneficial containers (like list, vector, map, etc.), a large number of algorithms (such as sort, find, count) among other utilities. They developed by *generic programming* approach [4], which means containers are defined as class templates and many algorithms can be implemented as function templates.

Functor objects make STL more flexible as they enable the execution of user-defined code parts inside the library without significant overhead [5]. Basically, functors are usually simple classes with an `operator()`. Inside the library `operator()`s are called to execute user-defined code snippets. Functors can be used in various roles: they can define predicates when searching or counting, they can define comparison for sorting or, they can define operations to be executed on elements.

C++ STL is widely-used inasmuch as it is a very handy standard. It is very common in programming when a user defined compound types are stored in a container provided by the standard library. However these types usually not ready to use with the most feature of STL. We cannot put it into a set, because there is no suitable `operator<`. We cannot apply an algorithm `find` on them because the lack of `operator==`. The solution is either to define the proper operators for these types, or specialize the containers, or algorithms of the STL with functor objects.

While the functors allow the comprehensive customization of the STL, writing functors often unconfortable, verbose, and it interrupts the flow of the code. This leads the standard committee to introduce lambda functions [6] in C++0x [7], the new standard of C++. Lambda function can be defined in place of a function argument, thus it simplifies the customization of algorithms in STL. However lambda function cannot passed to those placed, where not a concrete value, but a type is required. Thus containers cannot be customized by lambda function.

For the most user defined types the comparisons are semantically simple, only one or two fields are playing role to compute them. We can consider two persons equal, when their ids are the same, or one person is lesser than the other when its name is alphabetically lesser. Writing the necessary operators, for these types are often boring and error prone.

In this paper we provide a library, which is able to generate these operators by the related fields. The programmer can tag the fields to tells the library, which fields are playing role to compute a given operator. That way, with a small modification of the user defined type, all the necessary operators and other beneficial member function will be generated by our library making the type suitable to stored in any kind of container, and applyable to any kind of algorithm of STL.

Our paper is organized as follows: in section II we present a motivation example. Section III presents an overview of our solution, and section IV details our implementation. A complex example is found in section V. We conclude our results in section VI.

II. Motivation

Let suppose we want to store a user defined type `Person` in a `set` of STL. The definition of `Person` can be seen below.

```
struct Person
{
    int id;
    std::string name;
    std::string address;
};
```

The set to provide a fast response for the question: whether it contains a given element, it applies a search tree as its underlying data structures. Therefore the elements of a set

must be less then comparable. However our type `Person` has no `operator<`, thus we need to define a functor to tell the set the way to compare of object with our user defined type. The most efficient way to compare two `Person` is to compare their `id`-s. See the code below:

```
struct Cmp
{
    bool operator()(const Person& l,
                    const Person& r)
    {
        return l.id < r.id;
    }
};
```

Then we have to declare the set in the following way:

```
set<Person, Cmp> s;
```

Our solution simplifies this process, as it generates an `operator<` for our user defined type. We just need a slight modification of a our type: tag the field id, and ask the library to generate the `operator<`, which compares `Persons` with their id.

```
struct Person : less<Person, 1>
{
    tag1_final(int) id;
    std::string name;
    std::string address;
};

set<Person> s;
```

With our solution there is no need for functor and the simple version of set can be used. This solution is detailed in section III.

III. Tagging the fields

To apply our solution in a user defined type first we need to tag those fields, which are playing role on the required functionalities. Macroes `tag1`, `tag2`, ..., `tagn-1`, `tagn_final` are used for tagging. For the last tagged field instead of macro `tagn`, macro `tagn_final` must be applied.

The second step is to inherit the class from the functionality templates. The first template argument is the type itself [8]. The other template arguments defines which tagged fields to be used for the given functionality. Tagged fields are applied to the functionality in the same order as the template arguments reference them.

For example the functionality `dump<T, 1, 2, 3>` generates a member function wich prints the fields of type `T`. The field tagged by one is printed first, then the field tagged by two, and at last the field tagged by three. However the member function generated by `dump<T, 3, 2, 1>` does the same but in reverse order.

As the functionalities are generated by preprocessor [9], [10] and template metaprograms [11], they can accept as many template arguments as many tagged fields are.

At the current phase of the development our library provide the functionalities above:

- *less:* generates an `operator<`, which lexicographically compares two objects of the given type: First, compares the fields denoted by the first integer template argument. If they are different, we found the result. Otherwise it compares the fields denoted by the next integer template argument, as far as a difference occured or the last pair of fields are reached.
- *less_or_equal:* generates an `operator<=`, which applies lexicographical comparison.
- *greater:* generates an `operator>`, which applies lexicographical comparison.
- *greater_or_equal:* generates an `operator>=`, which applies lexicographical comparison.
- *equal:* generates an `operator==`, which returns true if all the tagged field pairs are the same.
- *not_equal:* generates an `operator!=`, which returns true if one of the tagged field pairs are different.
- *dump:* generates a `dump` member function, which prints the tagged fields. The order of printing the fields is the same as the integer template arguments refer them.
- *ostr:* generates an `operator<<`, which allows to apply an object of a given type on an *ostream*. *(E.g.: we can print the object with `cout`.)* The order of placing tagged fields to the *ostream* is the same as the integer template arguments refer them.
- *istr:* generates an `operator>>`, which allows to apply an object of a given type on an *istream*. *(E.g.: we can read the object with `cin`.)* The order of placing tagged fields to the *ostream* is the same as the integer template arguments refer them.

In the example below, the `struct Person` has an `operator<`, which compares persons by their ids, an `operator==` that applying the `name` and the `address` fields on the equality comparison, and the dump member function, which prints the values of all the fields to the standard output.

```
struct Person : less<Person,1>,
                equal<Person,2,3>,
                dump<Person,1,2,3>
{
    tag1(int) id;
    tag2(std::string) name;
    tag3_final(std::string) address;
};
```

IV. The tagging mechanism in details

The key step of the implementation was that, allows the functionality objects to access the fields in a user defined type by tags. The tag macro first saves the type information of the tagged type in a typevector [12] provided by the boost:mpl

library [13], then wraps the tagged type. The constructor of the wrapper saves the memory address of the fields into a static global array. Each object has an own array, and the arrays are stored in a map, where the key is the memory address of the object. As the type of the fields can be different, the array stores their pointers as `void*`. However, the type information is saved, thus the `void*` pointers can be easily cast back into their original type. The casting mechanism is done automatically by the library, behind the scene.

The wrapper class has to behave the same as the wrapped type. For compound types it is easy. The wrapper derives them and inherits all their functionalities. However it is not possible to derive from primitive types. That case the wrapper contains the primitive type as field, and the proper constructors, assignment and conversion operators are taking care of transparency. Type traits and partially template specialization technique provides unified usage of that two cases. See the code below:

```
template<typename T,
         bool pod=boost::is_pod<T>::value>
struct tag : T
{
    //wrapper for user defined types
};

template<typename T>
struct tag<T, true>
{
    //wrapper of primitive types
};
```

The `is_pod` is a metafunction [11] of boost typetraits library [14], which value evaluated true if its template argument is primitive type. The wrapper class `tag` has two template argument. The first one is the type to be wrapped, and the second one is a boolean, which equals the result of `is_pod` metafunction as default value. There is a partially specialized version of class `tag` on the second template argument. If the type to be wrapped is primitive type, the `is_pod` returns true, and the template mechanism get the partially specialized version of `tag` to be instantiated. Otherwise the general one is chosen.

The functionalities are template classes which perform a simple computation on selected tagged fields. The first template argument is the user defined type, which needs the functionalities. The other template arguments are integers, denoting those tagged fields which are playing role on functionality. Their number can be variadic from one to the number of all tagged fields. The following pattern is used to handle the variadic templates (let us suppose there are three tagged fields, and the functionality is the equality check):

```
template<typename T,
         int t1,
         int t2 = -1,
         int t3 = -1>
struct equal : virtual base
{
    bool operator==(const T& rarg)
    {
        // equality is computed
        // by three tagged field
    }
};

template<typename T,
         int t1,
         int t2>
struct equal<T, t1, t2, -1> : virtual base
{
    bool operator==(const T& rarg)
    {
        // equality is computed
        // by two tagged field
    }
};

template<typename T,
         int t1>
struct equal<T, t1, -1, -1> : virtual base
{
    bool operator==(const T& rarg)
    {
        // equality is computed
        // by one tagged field
    }
};
```

This solution is also based on partially template specialization mechanism. If the class `equal` is instantiated by only one tag id, the template argument `t2` and `t3` has their default value -1. Thus the third version of equal is chosen. When two tagged id is specified, only the last template argument has the default value, thus, the second version of equal is selected. If all the tagged id is set, the general version of equal is instantiated. The functionality templates are generated by preprocessor and metaprograms.

V. Example

This section presents a complex example about the usage of our library. In the example we want to create `Persons` described in section II, by reading their properties from the standard input. We want to store them in a set, and we want to print them to the standard output. To do this the type `Person` must have the following operators: `operator<`, `operator<<` and `operator>>`, which will be generated by our library.

```
#include <set>
#include <iostream>

#include <tag.h>
```

```
struct Person :
    less<Person, 1>,
    istr<Person, 1, 2, 3>,
    ostr<Person, 1, 2, 3>
{
    tag1(int) id;
    tag2(std::string) name;
    tag3_final(std::string) address;
};

int main()
{
    std::set<Person> sp;
    Person p;

    //reading persons
    while(std::cin >> p)
    {
        sp.inserT(p);
    }

    //writing persons
    std::set<Person>::iterator it;
    for(it = sp.begin();
        it != sp.end();
        ++it)
    {
        std::cout << *it << std::endl;
    }

    return 0;
}
```

VI. Conclusion and future work

In this paper we pointed out that altough the Standard Template Library of the C++ programming language can be highly customized by functors, this solution often cumbersome and it interrupts the flow of the code. This is principally concerns those user defined types, which semantically suits the requirement of the given entity of STL, but syntactically some functionalities are different or missing.

Using our library the programmer do not need to write these functionalities manually. With a small modification of the user defined type, our library is able to generate them during the compilation process. Our library based on standard features of C++, thus no external compiler is needed to compile it. Using our library, the programmer can concentrate on the real problems, which improves the quality of the source code.

One major deficiency of our solution is that the library is not thread safe. Our most important future work is to enhance the library to support parallelizm. Besides we plan to extinguish the implementation related restrictions of our library, and we would like to extend the functionalities that our library provides.

Acknowledgment

The Research is supported by the European Union and co-financed by the European Social Fund (grant agreement no. TAMOP 4.2.1./B-09/1/KMR-2010-0003)

References

[1] S. Meyers, *Effective STL*. Reading, MA: Addison-Wesley, 2001.
[2] N. Pataki, Z. Szűgyi, and G. Dévai, "C++ Standard Template Library in a safer way," in *Proc. of the Workshop on Generative Technologies (WGT 2010)*.
[3] "Programming languages C++," 2003, ISO/IEC 14882.
[4] A. Alexandrescu, *Modern C++ Design*. Reading, MA: Addison-Wesley, 2001.
[5] N. Pataki, "Advanced functor framework for c++ standard template library," *Studia Universitatis Babeş-Bolyai, Informatica*, vol. LVI(1), pp. 99–113, 2011.
[6] J. Järvi and J. Freeman, "C++ lambda expressions and closures," *Sci. Comput. Program.*, vol. 75, pp. 762–772, September 2010.
[7] B. Stroustrup, "Evolving a language in and for the real world: C++ 1991-2006," in *Proc. of the third ACM SIGPLAN conference on History of programming languages*, ser. HOPL III. New York, NY, USA: ACM, 2007, pp. 4–1–4–59.
[8] Z. Szűgyi and P. Norbert, "A more efficient and type-safe version of fastflow," in *Proc. of the Workshop on Generative Technologies (WGT 2011)*.
[9] B. Karlsson, *Beyond the C++ Standard Library, An Introduction to Boost*. Reading, MA: Addison-Wesley, 2005.
[10] [Online]. Available: http://www.boost.org/doc/libs/1_47_0/libs/ preprocessor/doc/index.html
[11] D. Abrahams and A. Gurtovoy, *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond*. Addison-Wesley.
[12] [Online]. Available: http://live.boost.org/doc/libs/1_47_0/libs/mpl/doc/refmanual/vector.html
[13] [Online]. Available: http://live.boost.org/doc/libs/1_47_0/libs/mpl/doc/refmanual.html
[14] [Online]. Available: http://www.boost.org/doc/libs/1_47_0/libs/type_traits/doc/html/index.html

Haskell Language from the Perspective of Compositional Programming

Vadim Vinnik, Tetiana Parfirova
Kiev National Taras Shevchenko University
Faculty of Cybernetics
Kiev, Ukraine
Email: (vadim.vinnik—tetiana.parfirova)@gmail.com

***Abstract*—Important features of Haskell language (monads and their binding operations) are compared to the conceptual basics of compositional programming – a paradigm in programming theory. On several examples it is shown that Haskell principles are in accordance with those of the compositional approach. It is stated that any well designed and successful conceptual design in programming should be based on principles already described in compositional programming. However, the later is still not widely used by authors in the world. Deliberate use of principles that are already discovered and described would be better than rediscovering them spontaneously in each design.**

***Index Terms*—Functional Programming, Semantics, Philosophical Foundations, Compositional Programming.**

I. Basics of the compositional platform

Goals and foundations of the compositional approach (CP) and its first results were published in late 1970s [1], [2]. The author of this conception stated that it is not inventing yet another *good programming style* (in addition to dozens of existing ones) but revealing the essence and deepest foundations of programming as such [2, p. 3]. Numerous subsequent papers were devoted to purely algebraic properties of compositions, applications of CP to various concrete areas of programming; other papers describe topics of entirely different level of abstraction: methodological, epistemological, ontological foundations of CP [6]. CP in its initial form together with based on it explicative programming, descriptology and entity-essence platform constitute an elaborate unified, integrative doctrine. Its core is a specific methodology, a philosophical approach to a subject (be it a programming task, a programming language, paradigm, a program, data object etc.) – all other aspects of CP (for example, mathematical models based on nominated functions) are derivatives of this approach.

At the same time, the extensive collection of results achieved in CP is obviously disregarded not only in everyday programmers' work (ignoring, sometimes even intentional, theoretical models and methods of theoretical computer science is, unfortunately, typical for low-skilled programmers and, moreover, is supported by market conditions) but also in designing new languages, frameworks, paradigms of programming – i.e. in areas where it is impossible to produce results without a sound methodology, and where a good theory is, in fact, the most practical tool.

However, in that extent in which CP (together with built-over platforms) achieves its goal (adequate, revealing deep foundations explication of its subject), any well well thought-out and successfully implemented large project should certainly use the same methods and pronciples as established in CP. The key point is, whether these methods and pronciples are used deliberately or spontaneously. In the second case, authors of numerous projects have to re-discover independently the principles and methods that are already well researched in CP. Indeed, it would be better to spare these efforts.

In this paper, authors attempt to state that the conceptual basis of rapidly achieving fame pure functional language Haskell [3] is mainly in accordance with CP. Since there are no evidences of using CP by designers of Haskell, this should be regarded as an example of spontaneous rediscovery of compositional and descriptological aspects.

It sould be noticed that one of the early works on CP was devoted to comparative analysis and revealing differences between CP and functional programming, where the later was represented by J. Backus' FP language [4]. The conclusion of that article is that Backus' FP should be regarded as a special case of CP: FP-style can be without perversions expressed by means of CP but not vice versa. The functional paradigm however has substantially evolved (in particular, FP language cannot be regardes as its representative), and this evolution eliminated some of defects mentioned in the article mentioned above.

Except this, the authors say in [4] that their "comparative analysis of CP and FP deals only with using naming relations and its consequences for program development and modification" (p. 27) and add that there could be indeed other criteria for comparing programming styles. Here we concentrate on the way that methodological principles of CP formulated after the mentioned article become aparent in a modern fuctional language.

II. Monads as Haskell's characteristic feature

Principles of Haskell design could be divided into two parts: functional proper and monadic ones. The first part consists of the idea of functional style itself together with supplying it features (typical, though in different extent, to all functional languages: FP, Lisp, Refal, Hope, Miranda, ML, F# and others) such as lazy computations, unacceptance of destructive assignments, referential transparency, carring, using high order functions etc. It would be probably difficult

to add something really new to this class of properly properly functional pronciples, there could only be more or less successful implementations.

The monadic part however defines Haskell's distinctive features. Indeed, there is some shade of monads in other functional languages: Scheme, Perl, F# but it was Haskell that first introduced monads, Haskell has the most comprehensive and mathematically pure support of monads. Except this, one can design programs in other languages with or without monads that play just a seconrady role, whereas in Haskell monads are intrinsic to the language and define its essence.

The notion of monad is polymorphic and has many sides, it is hard to give a short and exhaustive description, but there are, in general, two most significant aspects.

First, a monad (i.e. an object of a monadic type) acts as a container that stores values of some simpler types; number of components, their types and storing mechanism depend on a particular monad. For example, a list is a container that stores 0 or more values of one type rganised linearly; the type Maybe stores 0 or 1 value of certain type; the type Either has two slots for values of two types but only one slot can be used at each moment in time.

Second, a monad is an entity that encapsulates computations in an environment (in particular, actions with side effects such as input-output) allowing to embed them into purely functional programs, and to combine such computations into chains. From this perspective, a monadic object could play the role of a statement (in imperative programming sense).

Each particular monadic type shows these two properties in different extent. Lists, for example, show the first property and do not show the second one; IO monad is the opposite example. Let us consider two expressions: `[s]` where `s::String`, and `getLine`. The first expression has the type `[String]` which means a string wrapped into a monadic type List; the second expression's type is `IO String` – a string wrapped into a monadic type IO. One can see that there is no difference in the form of these two types but their substance differs radically. The object `[s]` does not represent any computation or action, does not prescribe to do something – it just stores a value as the only element of a list. Object `getLine` though does not store any value known in advance but instead encapsulates an action that reads a string from the input stream – and after that (speaking with a certain bit of conventionality) starts to store that value inside.

It is possible to reconcile these two faces of monads and combine them into a single notion: objects-containers like `[s]` could be regarded as trivial computations (all computation is limited to taking an already known value), and objects-actions like `getLine` could be considered as trivial containers (the storing mechanism is very simple because there is only one stored value but the way of obtaining this value is non-trivial because involves a non-deterministic or side-effects-prone input-output operation). Indeed, the language makes it possible to compose more complex monads having both roles non-trivial.

III. Monads as an intensional objects

It is noticed in [5] that among program entities (of which data objects, statements, types, programs etc. are just special cases) there could be selected *means for constructing entities* and, moreover, *means of application of constructing means*; the programming itself is described as a step-by-step process of applying (through certain application means) construction means to other entities. The difference between these three roles of entities is not extensional but intensional: it is not a property intrinsic to every entity but a kind of role it plays in certain situation. Each entity can in different contexts play any of these three roles. Therefore, programming as an intellectual process involves moving entities between the three levels.

It should be noticed that this pronciple is supported in functional programming at least by the fact that a function (that is usually a mean of constructing objects – its values) can be an argument or a value of other function (as ordinary objects). Moreover, a high-order function that has some function f as its argument in some way applies f to some values and, therefore, act as a mean of application of a construction mean f.

It is traditionally thought that the simplest mean of application of a construction mean to an object is a functional application – an operation that maps a pair (f, x), where f is a function and x is a value from its domain, to an object y that is the value of f on the argument x. However, it is stressed out in paper on CP that means for applying functions to data are not exhausted by classical applications mentioned above but, depending on the abstraction type selected for considering onject, can complex and multiform as much as is wished [6, p. 38, 47], and their diversity is not limited a priori. In the article mentioned above an entity that determines a way of applying a function to its arguments is called a driveration, so do we.

In Haskell, meanwhile, the binding operation `>>=` (the most important operation in the definition of monads) [3, p. 216, 223] is nothing but a driveration that controls how the function in its right-hand operand is applied, and how values obtained from such spplications are combined and wrapped together to a new monadic object.

Let us consider a function

```
f c = do x <- c; return (x+1).
```

Its type is

```
(Num b, Monad m) => m b -> m b.
```

In other words, provoded that `b` is a numerical type, and `m` is a polymorphic monadic type, `f` is a function that maps monads wrapping values of type `b` to monads of the same type.

Consider how this function is applied to two different monads: `Just 1` and `[1..5]`. The first object belongs to a monadic type `Maybe` (it either wraps one value or is empty), the second one is a list of 5 elements.

In the first case, the binding operation `>>=`, as it is defined for `Maybe`, discovers that the monadic object is not empty (i.e. contains some value) and, therefore, it can continue computations. The value 1 becomes unwrapped from the

monad, incremented, and the result is wrapped into a monad again. As a consequence, here binding acts as a conditional operation controlling application of `f`.

In the second case, the binding operation unwraps the values from the list one by one and applies `f` to each of them. Result of each application is wrapped into a list, after that all lists are concatenated. Here binding operation acts as a loop controlling application of `f`.

From this example, one can see that each monadic object plays not only a container's role but also a role of a driveration controlling a way how a function is applied to elements of that container. In other words, a monad has not only extensional (the values it keeps) but also an intensional – an abstraction type on which that values should be considered according to a respective driveration.

Now we can conclude that flexibility in moving objects from one intensional role to the other (which is important from the compositional point of view) is well supported in Haskell by wrapping and rewrapping the same value into different monads.

REFERENCES

[1] Redko V.N. "Compositions of Programs and Compositional Programming," Programmirovaine, 1978, Vol. 5, pp. 3–24 (in Russian).
[2] Redko V.N. "Foundations of Compositional Programming," Programmirovaine, 1979, Vol. 3, pp. 3–13 (in Russian).
[3] Dushkin R.V. Functional Programming in Haskell Language. – Moscow, DMK Press, 2007, 608 pp. (in Russian).
[4] Nikitchenko N.S., Redko V.N. "Compositional and Functional Programming: a Comparative Analysis," Programmirovaine, 1985, Vol. 2, pp. 15–28 (in Russian).
[5] Redko V.N. "Explicative Programming: Retrospectives and Perspectives," in Proceedings of the 1st International Comference UkrProg'98, Kiev, 1998, pp. 3–24 (in Russian).
[6] Redko V.N. "Foundations of Programmology," Kibernetika i sistemny analiz, 2000, Vol. 1, pp. 35–57 (in Russian).

Introduction to Domain Analysis of Web User Interfaces

Michaela Bačíková[1], Jaroslav Porubän[2], Dominik Lakatoš[3]
Department of Computers and Informatics
Technical University of Košice
Košice, Slovakia
michaela.bacikova@tuke.sk[1], jaroslav.poruban@tuke.sk[2], dominik.lakatos@tuke.sk[3]

Abstract—**Graphical user interface (GUI) is the most important part of application, with which users interact directly. It should therefore be implemented in the best way with respect to understandability. Based on this assumption we decided to analyze user interfaces for domain specific information. In our previous works, we strived for a classical desktop GUI analysis and we used a component-based approach. It was done through defining a simple domain-specific language (DSL). Now we would like to continue in this work, but in the field of web user interfaces. The internet provides more promising amount of resources. In this work, we propose taxonomy of web components based on their domain-specific information and we outline our method for extracting this information. In the future, we would like to create a system that is capable of automatically analyzing web UIs. Instead of a DSL we would like to generate ontologies.**

Keywords-ontology; graphical user interface; domain-specific language; domain driven component-based analysis; Java; ontology learning; web user interfaces

I. INTRODUCTION

A. Analysis of user interfaces

The question of creating a user interface (UI) has a clear motivation: system that communicates with a user leads to a dialog via UI and therefore it is necessary to construct it – undoubtedly. Sometimes it is hard to separate an analysis and a design phase of software development. An example is when user is a part of development process where the software is incrementally created and in every cycle it is analyzed by developers together with the user.

We however focus on analysis, which is not aimed primarily to create or modify an existing system (our analysis is not the primary feedback for design). In this case it is not required of us to be a part of design and modifications of the system and our conclusions will be useful beyond the existing system. We are conducting analysis for other needs than for design.

This analysis can be used in several fields:

Existing systems are (in most cases) the best existing formal description of a given domain. Based on this assumption, it is possible to **initially construct a (semi-)formal description of the domain** (a language or an ontology or a tool support for creating sentences) based on UI analysis.

UI usability evaluation – comparing two or more systems (or a new and an old version of a single system) from the perspective of usability, system evaluation by users/automatically.

Generating of user guides

Note: The form of analysis outputs will probably depend on the intended use.

We decided to analyze web user interfaces mainly for the fact, that the Internet is a good source of analysis inputs - one could say: infinite.

B. Analysis of existing systems

Currently, there are three types of analysis of existing systems:

a) *on a knowledge base level (database)*
b) *on a source code level*
c) *on a presentation level (GUI)*

Because user has any access nor to a) or b), these two levels do not need to reflect the domain precisely. The terms used could be written in another language, expressed in shortcuts or there could be any other language barriers. A generic approach to programming prevents expressing the domain (the generic terms are not usable for a specific domain). In a) there is an absence of descriptions of domain processes (apart from the procedures, but we include them into source codes). By contrast, although the source codes represent the domain processes, the source code analysis is very complicated.

C. Analysis of web user interfaces

In our previous work (see *III.A.*) we have tried to automatically formalize interfaces into a form of a very simple domain-specific language (DSL). To design a language, that would provide adequate expressiveness of concepts, their properties, relations with other concepts and hierarchy of interface components, is difficult. Therefore we decided to use an existing solution meeting all these features – ontologies. Existing articles provide many definitions of ontology [10], [12], [16], [17]. We rather use one of them that fully meets our goal than create another one: ***Ontology*** *is an explicit representation of concepts of some domain of interest, with their characteristics and their relationships.*

The benefits of our approach would be a contribution to Semantic Web by generating ontologies automatically. ***Semantic Web*** is an extension of the current web in which the data are described in a way that both people and computers

understand. It should facilitate cooperation between people and machines. Central part of the Semantic Web are ontologies. Vision is, software agents will search the web for our requirements and based on them, they offer us solutions to our problems. Manual creation of ontologies is, however, a difficult and tedious work although several tools for creating ontologies exist like Protégé [24]. Therefore there is a need for an automatic creation of ontologies, referred to as an ***ontology learning***.

This articles' goal is to further investigate the extraction process proposed in our previous work (see *III.A.*) in terms of Web interfaces and ontologies. Contribution of this article would be the proposed taxonomy of GUI components and their preliminary analysis related to domain-specific information. So far, there has not yet been published any such classification *with emphasis on defining a GUI metamodel for extracting domain-specific information*. This classification will serve our further research in this area and it is a basis for creating our automatic ontology learning system.

D. Tasks and Goals

From our goal, to create a system that would automatically formalize a GUI to generate ontologies, the following basic goals, targets and assumptions have emerged:

Goal 1: Explore the possibility of extracting ontological concepts from different types of interfaces using a component-based analysis.
Assumption 2: The GUI must be made of components.

Note: Assumpt. 1 was defined in our previous work (see *III.A.*).
Note: Whereas the Web is an endless knowledge base, we will focus primarily on web documents.

Goal 3: Determine whether the analysis of a web interface (or its parts) is possible.
Goal 4: Outline the ontology learning method.
Subgoal 4.1: Identify potential problems related to this approach.
Subgoal 4.2: Suggest other mechanisms (automatic or manual) to improve the outcome.

The paper is further organized as follows: Section II divides GUIs into several groups according to their form and describes possibilities of formalization for each type. We will try to outline our previous work and the state of the art in the field of ontology learning in section III. In section IV we will outline the proposed method for automatic ontology extraction and perform an extensive analysis of existing web components and their meaning as domain specific units which will serve our further research. We also propose new taxonomy for UI components tailored for our approach. Finally section V will provide an insight into our future research and a section VI is a conclusion.

II. Graphical User Interfaces

We can divide GUIs into five basic groups with respect to their form. We will describe them (and possibilities of their formalization using a component-based approach) in the following subsections.

A. Console UIs

We write "UIs", not "GUIs" because these are not fully "graphical" UIs. They are obsolete text interfaces that use no graphics. If all text items displayed in the interface program are stored in variables, it is easy to formalize such an interface. But such formalization is rather an analysis on a source code level. Although there are approaches doing this type of analysis like [20] (this work deals with transformation of obsolete UIs into WIMP GUIs). But it is a research from 1997, so we could say it's obsolete. Console UIs are rarely used today therefore we will not deal with this type of interfaces further.

B. WIMP GUIs

The "WIMP" abbreviation stands for "Windows, Icons, Mouse, Pointers" and represents standard desktop applications. Formalization of such interfaces depends on the type of programming language and the programming style. If governed by guidelines (for example [19]), analysis possibilities grow wider. Analysis of this type of interfaces was a center of our previous work, described in III.A.

C. Web GUIs

In the context of this article we refer to web documents as *web applications*, for we are dealing with interactive web UIs. Interacting with the interface is either (in HTML documents) by clicking the network links, or (in interactive web applications) similar to WIMP applications – but with content remaining in the web browser (no windows). In this article we will further deal with this type of interfaces.

D. Mobile GUIs

Mobile devices use a completely different and very much simpler type of user interfaces mainly because of their limited operability. There are two types – non-touchscreens and touchscreens. Mobile interfaces (in most cases) consist of *Screens* that can be pushed to (or popped off) the device screen. Every Screen component has its *title* and *content* (in form of graphical components). Domain analysis of such screens should be easy, because they clearly form a hierarchy of concepts. But the process of attaching an analyzer program to a mobile device or mobile device simulator could be complicated. Therefore for now we will not deal with these and we leave them for further research.

E. Other GUIs

This group includes interface of applications that do not fit into the previous four groups, for example Java Web Start applications, applets, flash, touch screen applications (iPad), etc. We will not deal with these types for now.

III. State of the Art

A. Previous Work

Our previous research has addressed the domain-driven formalization of WIMP GUIs [21], [22], [23]. A formalization program was implemented that produced output in form of a DSL named GUIIL (GUI Interaction Language) designed specifically for describing UIs in a simple form. The created

system was capable of extracting terms based on user interaction with GUI. We used Java open source applications for performing experiments. Then after the extraction our system could automatically "replay" the same commands on the application based on the GUIIL file as it would be performed by a user. We used component-based analysis to implement this method.

The idea of the research is that the commands in GUIIL file were *domain identifiers of GUI components*. For example, if there was a button named "Next", its domain identifier was "Next". When in replay mode, the system searched for appropriate component based on its identifier and performed appropriate action on it (for a button, clicking was pre-defined) like it would be performed by a user. The action was not specified in the input file, just the identifier.

Our previous works also contain a detailed description of concepts, features and relations that can be automatically derived and extracted from WIMP GUIs and this was used in the system to derive an action from a component type. We created DSLs to define GUIs, but the types of features in the interfaces rather define ontology than a DSL. We believe that *by comparing the application interface metamodel with model of a concrete application it is possible to automatically derive and generate ontologies*.

In this article we would like to explore the possibilities of this assumption, to analyze components, their features, relations, and their membership in a hierarchy of concepts derivable by their definitions, structure and location. We will focus on the Web UIs. Concerning ***Assumption 1*** defined earlier, the web GUIs are made of components in a form of HTML tags that can be easily formalized using XML or HTML parsers. Thus it's possible to use the component-based approach to analyze Web GUIs.

B. Related work

Vision of Semantic Web requires that web documents had semantics defined. The current web however provides documents to read, but not to understand – web agents don't know anything about their semantics. Ontologies are used to resolve this problem. And the benefit from them is already in a process of utilization. For example in [18] a predefined ontology was used to extract comments from websites to obtain user feedback on some business products, technologies or information systems.

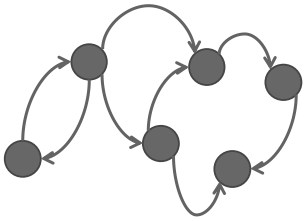


Figure 1. Initial vision of the Web – a net of simple documents/sources

At present therefore many approaches are targeted to ontology learning. Several methodologies for building ontologies exist, such as OTK, METHONTOLOGY or DILIGENT, but they target ontology engineers and not machines [10]. Many methods and different sources of analysis are used to generate ontologies automatically. Among the obsolete and less effective form we would include NLP (Natural Language Processing), used for example in [5]. These approaches are however complicated and their perception Internet as a net of simple documents is obsolete (Fig. 1.). The Internet has rather transformed into a web of (dynamic) interactive applications (see Fig. 2.).

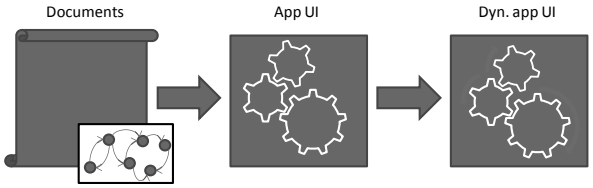


Figure 2. Transformation of web documents and their perception

A little bit closer to this perception are analyses of concrete web structures (tables) or document fragments or just the basic attributes (title, head) - see methods ix) and x) below. But these are just little parts of a whole web UI.

Results are almost always combined with a manual controlling and completing by a human and as an additional input, there's almost always some general ontology present (a "*core ontology*") serving as a "guideline" for creating new ontologies. Different methods are used to generate ontologies:

i) clustering of terms [7], [13], [15],
ii) pattern matching [5], [13], [14], [15],
iii) heuristic rules [5], [14], [15],
iv) machine learning [1], [12],
v) neural networks, web agents, visualizations [15],
vi) transformations from obsolete schemes [14],
vii) merging or segmentation of existing ontologies [8], [13],
viii) using fuzzy logic to generate a fuzzy ontology, which can deal with vague terms such as FFCA method ([2] and [16]) or FOGA method ([6]),
ix) analysis of web table structures [4], [5], [17],
x) analysis of fragments of websites [3].

A condition for creating a good ontology is to use many sources as an input to analysis - *structured*, *non-structured* or *semi-structured* - and to use a combination of many methods [10]. Therefore as an additional mechanism for identifying different types of relationships (e.g. mutual exclusivity, hierarchical relations), a web dictionary WordNet ([4], [9], [11] and [13]) or other web dictionaries or databases available on the Web are used.

A state of the art from 2007 can be found in [10].

IV. Proposed Method for Automatic Ontology Extraction

We would like to design such a method which would not require human control at runtime and it would require just a minimal check at the end of the process. We also want to analyze websites in a simple, transparent way to enable extensibility and focus only on part that creates GUIs – because the presentation layer of an application is the first thing that a user sees and it should be programmed in such a manner that a general user would understand its language. This presumption allows us to analyze web documents as UIs. Although we will

use an existing general ontology as an input, this ontology will be created only once by us and will no longer be changed (only when programmer would like to extent the programming). It will define GUI web documents on a metalevel and will serve as a "guideline" for creating new ontologies. In the following sections we will try to describe how we define the characteristics of some Web graphical user interface components. **Based on these characteristics we will create the algorithm for GUI formalization.**

A. Assumptions and classification

In our previous work we were conducting analysis based on following assumptions:

i. GUI is made of components.
ii. These components are domain-specific units – they represent **terms** of a GUI language.
iii. For these terms, we can define **properties**, **rules**, there are some **relations** between them and they are creating a **hierarchy of terms**.
iv. We can **identify**, **extract** and **process** them.

We will continue to use these assumptions when performing analysis of web components. In this work we use a different classification than in our previous research. We believe that this proposed taxonomy will better serve the web interface automatic analysis algorithm:

i. Components giving metainformation about their content
ii. Components defining a structure or a hierarchy
iii. Components defining relations between items of their content
iv. Functional components
v. Textual and information components
vi. Formatters and separators

Now we will present the result of conducted analysis using this classification.

B. The analysis of Web GUI term extraction possibilities

Analysis of web components and their classification to the defined groups describes TABLE I. In the first column there are names of components represented by HTML tags. In the second, there are groups of the taxonomy defined in IV.A (as i, ii, iii, iv, v, vi). In the third column we give an example of use. The code examples were selected from www.w3schools.org. And finally, in the last column we describe each type of component and domain specific information that it defines. This is important for designing the GUI analysis algorithm. For some components (or their parts) it is possible to extract many types of information, therefore we included these components into several groups.

Note: because of limited space, we have chosen only the most commonly used components and such, that can offer the most important domain-specific information defining the various concepts, structures, hierarchy and relations between future ontology concepts.

TABLE I. ANALYSIS OF COMPONENTS OF A WEB GUI

HTML Tag	Group	Example of use	Domain-specific information, which the component defines
<a>	ii, iv	 Click me! 	Defines hierarchy or web of web documents.
<abbr>, <acronym>	i	The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.	Defines meaning of a shortcut / acronym – metainformation.
<address>	i	<address> Written by W3Schools.com Email us Address: Box 564, Disneyland </address>	Define type of their content – the page creator address – metainformation.
<button>	iii	<button type="button"> Click Me! </button>	Customizable push button. Functional component performing some action and changing state of application.
<form>	i, iii, iv, v	<form> <legend>Person</legend> <label for="male"> Male </label> <input type="radio" name="sex" id="male" /> <label for="female"> Female </label> <input type="radio" name="sex" id="female"/> </form>	We can extract many domain specific information of a form: - If the form has a <*legend*> tag (like in this case "Person") or a *name* attribute specified, it specifies (sub)domain of the form content. - The *input* components are of many types • *submit* - submits information to server, • *radio* - terms expressed by radio buttons are mutually exclusive, • *text* - represents textual information, its properties and limitation, • *password* - represents encoded information, password, • *checkbox* - terms expressed by checkboxes are not mutually exclusive. - For description of the input fields, the <*label*> tag with a *for* attribute is used. This specifies a meta information about the input component. Note: a work similar to ours is already in utilization by existing web browsers, for example identifying and remembering user password – the fields with username and password are automatically identified and their content stored in browser's memory.

HTML Tag	Group	Example of use	Domain-specific information, which the component defines
<select>	ii, iii	<select name="cars"> <option value="volvo"> Volvo </option> <option value="mercedes"> Mercedes </option> </select>	Selection list item. - the *name* or *id* attribute can tell us about the domain of selection's items - the *items* in the selection list represent mutually exclusive terms Selection list item can be organized also hierarchically, using an <optgroup> item. From such structure we can directly extract the whole hierarchy of terms.
<dl>	ii, iii	<dl> <dt>Coffee</dt> <dd>- black hot drink</dd> <dt>Milk</dt> <dd>- white cold drink</dd> </dl> <ul id="menu"> MenuItem1 	Definition list. The <*dt*> tag gives items of the definition list. Of them we can tell, that they belong to a same subdomain. The <*dd*> tag gives description of a definition list item. Very similar tags are <*ul*> (unordered list) and <*ol*> (ordered list) and deprecated <*dir*> (directory list), all with <*li*> items, but they do not have the description tags (dd). Some HTML creators use these lists to create a menu of a webpage (see the second example with *menu* attribute). From such structures we can directly extract page menu, e.g. we get the hierarchy of web pages. There is also a deprecated tag named <*menu*>, that looks exactly like unsorted list. This better describes menu content that can be extracted.
, , <dfn>, <code>, <samp>, <kbd>, <var>, <cite>	i	Emphasized text Strong text <dfn>Definition term</dfn> <code>Computer code</code> <samp>Sample computer code</samp> <kbd>Keyboard text</kbd> <var>Variable</var> <cite>Citation</cite>	Formatting tags give metainformation about their content. Commonly when writing some text, the important parts of documents, definitions, or key words, are written using italic or bold format. For tags like <*em*> and <*strong*> we therefore can assume, that their content is something important. About the other tags, meaning of their metainformation is obvious when looking at their name or sample codes.
<table>, <caption>, <th>, <tr>, <td>	i, ii	<table border="1"> <tr> <th>Month</th> <th>Savings</th> </tr> <tr> <td>January</td> <td>$100</td> </tr> </table>	Tags defining tabular structures. From <*caption*> tag we can get table (sub)domain, e.g. metainformation about table content. From <*th*> we can determine (sub)domain of columns content. This way we can create a concept hierarchy. From cell properties we can derive properties of ontology concepts. Similar work was done in [4], [5], [17], mostly using NLP, regular expressions, type recognizers and clustering. We would rather keep it simple when dealing with tables so the hierarchy of a whole webpage wouldn't go into too much detail, because then it would be very difficult to maintain such a big ontology.
<textarea>	v	<textarea rows="2" cols="20"> At W3Schools you will find all the Web-building tutorials you need, from basic HTML to advanced XML, SQL, ASP, and PHP. </textarea>	Text area. From text components we can get domain specific information like: - type of data, - maximal length, - content, - constraints, - purpose from the <*label*> component, etc.
<tt>,, <b>,<big>, <small>, <h1> - <h6> etc.	vi	 This is some italic text. 	Formatters and separators. Besides separating different types of content from each other they do not contain any relevant domain-specific information. Formatting tags serve similar purpose as , , etc.
<head>	i	<head> <title>This is title</title> <head>	From web page *head* we can extract page's *name*, e.g. name of web page's domain. Using a <*meta*> tag the programmer defines metadata about the website.
<img>, <object>, <map>	i	 <object width="400" height="400"> <param name="movie" value= "/flash/helloworld.swf"> </object>	Component of type <*img*> contains a picture. An object can contain image, audio, video, Java applet, ActiveX, PDF or Flash. Tag <*map*> contains a clickable image map. As a domain-specific information we can consider the object type, e.g. a "media or interactive content" – a metainformation.

V. Further Research (The Proposed Method)

As written in the above sections, we would like to continue our research in the field of domain driven GUI analysis and there's much work to be done. We will continue to focus on web interfaces, because this area provides promising amount of resources.

Proposed analysis of components must be extended to cover the most of web GUI components and to explain in detail, what domain specific information can be extracted for the algorithm to be defined. A formalization method of each type of component must be defined. Based on this analysis, the general (core) ontology must be defined and created to serve the creation of domain-specific ontologies. The method outlined in this article must be finalized and implemented into a system capable of automatic ontology learning. ***Subgoal 4.1*** and ***Subgoal 4.2*** defined in I.B. must be fulfilled. To better the results WordNet or other dictionaries can be used. Experiments must be done with the implemented system and results evaluated and this information should help fulfilling the subgoals 4.1 and 4.2.

VI. Conclusion

This paper serves as an introduction to domain analysis of web GUIs. It is a basis for our further research. In this work we proposed a new taxonomy for web components serving domain driven component-based analysis. We described our method and tried to make an extensive analysis of existing HTML components to serve the design of our algorithm. Based on this analysis we will try to implement a tool which is capable of automatic generation of ontologies and by this it will contribute to the growth of the Semantic Web.

Acknowledgment

This work was supported by VEGA Grant No. 1/0305/11 – Co-evolution of the artifacts written in domain-specific languages driven by language evolution.

References

[1] B. Omelayenko, "Learning of ontologies for the web: the analysis of existent approaches," in In Proceedings of the International Workshop on Web Dynamics, 2001.

[2] T. T. Quan, S. C. Hui, A. Fong, and T. H. Cao, "Automatic generation of ontology for scholarly semantic web," The Semantic Web ″U ISWC 2004, pp. 726–740, 2004.

[3] T.-L. Wong, W. Lam, and E. Chen, "Automatic domain ontology generation from web sites," J. Integr. Des. Process Sci., vol. 9, pp. 29–38, July 2005.

[4] A. Pivk, "Automatic ontology generation from web tabular structures," AI Communications, vol. 19, p. 2006, 2005.

[5] Y. A. Tijerino, D. W. Embley, D. W. Lonsdale, Y. Ding, and G. Nagy, "Towards ontology generation from tables," World Wide Web, vol. 8, pp. 261–285, September 2005.

[6] Q. T. Tho, S. C. Hui, A. C. M. Fong, and T. H. Cao, "Automatic fuzzy ontology generation for semantic web," IEEE Trans. on Knowl. and Data Eng., vol. 18, pp. 842–856, June 2006.

[7] S.-h. Sie and J.-h. Yeh, "Automatic ontology generation using schema information," in Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI '06, (Washington, DC, USA), pp. 526–531, IEEE Computer Society, 2006.

[8] J. Seidenberg and A. Rector, "Web ontology segmentation: analysis, classification and use," in Proceedings of the 15th international conference on World Wide Web, WWW '06, (New York, NY, USA), pp. 13–22, ACM, 2006.

[9] Y. J. An, J. Geller, Y.-T. Wu, and S. A. Chun, "Automatic generation of ontology from the deep web," dexa, vol. 0, pp. 470– 474, 2007.

[10] I. Bedini and B. Nguyen, "Automatic ontology generation: State of the art," tech. rep., University of Versailles Technical report, 12 2007.

[11] Y. Ding, D. Lonsdale, D. W. Embley, and L. Xu, "Generating ontologies via language components and ontology reuse," in In Proceedings of 12th International Conference on Applications of Natural Language to Information Systems (NLDBŠ07, 2007.

[12] J. Shim and H. Lee, "Automatic ontology generation using extended search keywords," in Proceedings of the 2008 4th International Conference on Next Generation Web Services Practices, (Washington, DC, USA), pp. 97–100, IEEE Computer Society, 2008.

[13] H. Yang and J. Callan, "Ontology generation for large email collections," in Proceedings of the 2008 international conference on Digital government research, dg.o '08, pp. 254–261, Digital Government Society of North America, 2008.

[14] M. Wimmer, "A meta-framework for generating ontologies from legacy schemas," in Proceedings of the 2009 20th International Workshop on Database and Expert Systems Application, DEXA '09, (Washington, DC, USA), pp. 474–479, IEEE Computer Society, 2009.

[15] J. R. G. Pulido, S. B. F. Flores, R. C. M. Ramirez, and R. A. Diaz, "Eliciting ontology components from semantic specific-domain maps: Towards the next generation web," in Proceedings of the 2009 Latin American Web Congress (la-web 2009), LA-WEB '09, (Washington, DC, USA), pp. 224–229, IEEE Computer Society, 2009.

[16] W. Chen, Q. Yang, L. Zhu, and B. Wen, "Research on automatic fuzzy ontology generation from fuzzy context," in Proceedings of the 2009 Second International Conference on Intelligent Computation Technology and Automation - Volume 02, ICICTA '09, (Washington, DC, USA), pp. 764–767, IEEE Computer Society, 2009.

[17] X. Lei and R. Yong, "Ontology generation from web tables: A 1+1+n approach," in Proceedings of the 2010 International Forum on Information Technology and Applications – Volume 01, IFITA '10, (Washington, DC, USA), pp. 234–239, IEEE Computer Society, 2010.

[18] M. Neunerdt, B. Trevisan, T. C. Teixeira, R. Mathar, and E.- M. Jakobs, "Ontology-based corpus generation for web comment analysis," in ACM conference on Hypertext and hypermedia (HT 2011), (Eindhoven), 05 2011.

[19] Java Look & feel design guidelines, http://java.sun.com/products/jlf/ed2/book/, 2001

[20] Melody M. Moore and Spencer Rugaber. 1997. Domain Analysis for Transformational Reuse. In Proceedings of the Fourth Working Conference on Reverse Engineering (WCRE '97). IEEE Computer Society, Washington, DC, USA, pp. 156.

[21] Jaroslav Porubän, Michaela Kreutzová, "Definition of computer languages via user interfaces". In: Electrical Engineering and Informatics: Proceeding of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, Košice, Slovak Republic. 2010, pp. 53-57.

[22] Michaela Kreutzová, Jaroslav Porubän, "Automating User Actions on GUI: Defining a GUI Domain-Specific Language". In: CSE 2010: proceedings of International Scientific conference on Computer Science and Engineering: Stará Ľubovňa, Slovakia, 2010 pp. 60-67.

[23] Michaela Kreutzová, "Basis for GUI domain usability analysis: Formalization". In: SCYR 2011: 11th Scientific Conference of Young Researchers of Faculty of Electrical Engineering and Informatics Technical University of Košice: proceedings from conference, Herľany, Slovakia. 2011 pp. 243-246.

[24] Protégé, tool for creating and editing ontologies. http://protege.stanford.edu/, 2011.

Reverse Language Engineering: Program Analysis and Language Inference

Ján Kollár, Sergej Chodarev,
Emília Pietriková and Ľubomír Wassermann
Department of Computers and Informatics,
Faculty of Electrical Engineering and Informatics
Technical University of Košice, Slovak Republic
E-mail: {jan.kollar, sergej.chodarev,
emilia.pietrikova, lubomir.wassermann}@tuke.sk

Dejan Hrnčič and Marjan Mernik
Institute of Computer Science,
Faculty of Electrical Engineering and Computer Science
University of Maribor, Slovenia
Email: {dejan.hrncic, marjan.mernik}@uni-mb.si

***Abstract*—Abstraction is one of the most important concepts in computer science. This paper presents approach for analysis of abstraction based on program samples. Developed tool for analysis of Haskell programs from syntactic point of view is presented. Derivation trees of programs provided by the tool are used in presented approach for measurement of formal abstraction effect. We propose a method for automatised raise of abstraction level based on recognition of patterns in programs code. Language learning or grammatical inference is a technique for inferring a context-free grammar from a set of positive and/or negative samples. In this paper we also present a memetic algorithm for grammatical inference, called MAGIc. MAGIc is demonstrated on small part of Haskell language, data type declaration. MAGIc is able to infer correct context-free grammar from positive samples only.**

I. Introduction

Evolution of programming languages plays indispensable role in programming. The use of high-level programming languages was the most powerful stroke for software productivity, reliability and simplicity [1]. Programming languages are thought models they greatly influence the way of how problems are solved in programs. Expressive power of a language and its basic concepts have great impact on developed program – its structure and style [2]. In the result, language also influences practical characteristics of programs – their extensibility, reliability, efficiency, and much more.

Due to these reasons, programming languages became a popular research topic [3], [4], [5]. For proper understanding of a language, it is viable to analyze it upon its practical usage. One of the methods includes using samples of programs implemented in the language. Program samples provide valuable source of knowledge about the language. Furthermore, they provide a view on real usage of the language.

On the other hand, samples of programs can also be used to infer syntactic structure or grammar. This is useful in legacy programming languages, where language documentation is missing, or it can be used also in process of designing a new programming language (whether extending existent one or developing a new language). With automatically extracted grammars, language processing tools (e.g. parser) can be generated and used in further language design. Extracting grammars from samples of programming languages is part of grammatical inference field [6].

In Section II a grammar based samples analysis is presented and discused on example of Haskell language. Section III presents memetic algorithm for grammatical inference MAGIc and an example of inferring a context-free grammar for the part of Haskell language. The paper concludes with Section IV.

II. Grammar Based Samples Analysis

If grammar of the analyzed language is available, it is possible to use it within the analysis of programs samples. On the other hand, if grammar of the analyzed language is not available yet the language inference approach described in Section III should be applied first. Analysis of programs based on syntax of the language can provide a view on the real usage of the language.

Syntactical view of a program can be achieved using its derivation tree that reflects the whole process of syntax recognition of the program. Thus, the first step of analysis is to retrieve derivation trees of programs based on the grammar. Later, these trees can be inspected to retrieve various statistical data. For instance, it is possible to find out relative usage of the language elements in programs. Moreover, it is possible to recognize common program patterns as well.

By program patterns, we understand code fragments extracted from a set of sample programs that have equivalent syntactic, and hence, also semantic structure. Patterns can also contain parts that differ in each program. These parts are called syntactic variables.

In fact, program patterns represent syntactic structures repeatedly used in programs. They might be used within evolution of language as a basis for new language abstractions. After introduction of new abstraction based on a pattern, syntactic variables will become parameters of the abstraction.

A. Haskell Syntax Analysis

To realize the described conceptions, Haskell was chosen as an analyzed language. To gather needed information from Haskell programs, a set of tools was developed to get a proper

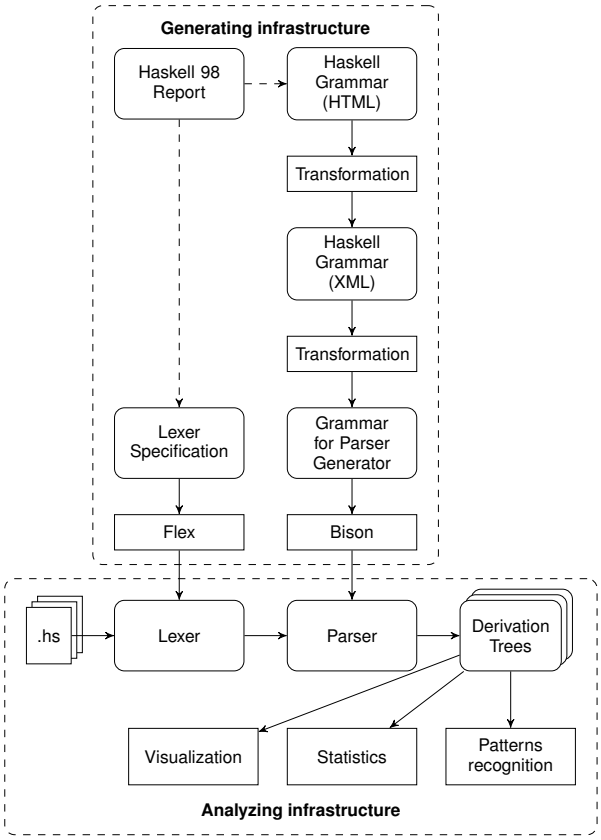


Fig. 1. Architecture of Haskell Syntax Analysis

knowledge about constructs used in analyzed programs. As a result of the program analysis, derivation tree is produced, consisting of used rules of the Haskell grammar [7]. Architecture of Haskell syntax analysis consists of two parts – *generating* and *analyzing infrastructure* (see Fig. 1).

The goal of generating infrastructure is to prepare tools used during the analysis of Haskell programs by the analyzing infrastructure. These tools (lexer, parser) have to conform to Haskell grammar to be able to process its programs.

The analyzing infrastructure in Fig. 1 contains lexer and parser of Haskell programs, intended for analysis of Haskell programs into lexical units, then processing them into derivation trees.

Derivation trees are produced in the XML format, subsequently visualized using the Graphviz tool [8] and further processed to retrieve statistical data on Haskell programs, and to recognize common language patterns.

1) Haskell 98 Syntax: Haskell is a general purpose, purely functional programming language that provides higher-order functions, non-strict semantics, static polymorphic typing, user-defined algebraic data types, pattern-matching, list comprehensions, a module system, a monadic I/O system, and a rich set of primitive data types, including lists, arrays, arbitrary and fixed precision integers, and floating-point numbers [9].

In [10], Haskell syntax is processed into HTML format. This form was chosen as a basis for development of the parser. It uses extended Backus-Naur Form.

To speed-up the development, it is necessary to use a parser generator tool. The choice of the appropriate parser generator depends on a class of the processed grammar. Haskell grammar contains several cases using the left recursion, so it has been treated as the LR type. Moreover, reduce/reduce conflicts may be found in the grammar, regarding several rules sharing the same right side. LALR parser can not choose the right nonterminal to reduce, therefore using GLR parser generator might be appropriate, being capable of parallel reduce of each nonterminal, trying to proceed with multiple possibilities. Because of its ability to generate a GLR parser, Bison [11] was chosen as parser generator.

Lexical analysis is then divided in two steps:

a) Traditional lexical analysis based on lexical grammar. Lexical analyzer is generated by the Flex tool [12].
b) Processing of program layout, where white space characters are analyzed following the algorithm specified in Haskell 98 Report [7] and replaced by tokens corresponding to braces and semicolons.

2) Grammar Transformation: To be able to process the grammar programmatically, HTML representation of Haskell 98 grammar was transformed into XML form, more suitable for further processing. XML grammar is then used to generate grammar specification in a format suitable for the parser generator. During this process, grammar rules need to be transformed from EBNF to BNF form accepted by Bison.

3) Grammar Ambiguity: Meanwhile the parsing, several ambiguities were detected. It was needed to modify rules `pat_i` and `exp_i` to avoid situation when parser was not able to join parallel parsings.

Ambiguity was also spotted in the rules `export`, `import`, `aexp`, and `alts`, where parser could not determine, which alternative to choose. The problem was solved by setting priority using the `%dprec` directive.

Another ambiguity is related to lambda (`\`) and conditional (`if-then-else`) expressions, where parser could not determine correctly, where the expression ends. According to Haskell Report [7] it is supposed to continue as far as possible. As a solution, the `%merge` directive was used with custom merge function, that given two possible subtrees as arguments decides which is correct one.

4) Fixity Resolution: Another problem, that needed to be solved, was a resolution of fixity and precedence of operators. Instead of defining separate grammar rules for all precedence levels of expressions (like it is done in Haskell 98 Report [7]), infix expressions are defined using one rule and fixity and precedence are resolved after parsing. Resolution is done according the algorithm specified in Haskell 2010 Report [13].

B. Code Statistics

Using developed tools, it was possible to compute some interesting statistics based on a set of about 300 Haskell

TABLE I
PROPORTION NUMBER OF 40 MOST FREQUENT SYMBOL OCCURRENCES

Symbol	Occurrence	Symbol	Occurrence
varid	0,093855	qvarop	0,015110
aexp	0,092660	gcon	0,014879
fexp	0,092660	atype	0,013402
exp_10	0,063059	varsym	0,012450
qvar	0,051154	qcon	0,011951
exp_i	0,049428	btype	0,011516
exp	0,044523	,	0,011309
var	0,037632	funlhs	0,010876
apat	0,033349	type	0,010080
conid	0,026202	]	0,008857
(	0,019259	[	0,008857
)	0,019259	integer	0,008808
=	0,018341	gtycon	0,007696
decl	0,017526	pat	0,007687
;	0,017277	string	0,005727
qop	0,016620	->	0,005454
topdecl	0,016484	}	0,004845
rhs	0,016164	{	0,004845
pat_i	0,016099	|	0,004296
pat_10	0,016099	qconop	0,003216

sample programs. As a result of program analysis, its derivation tree is provided according to the language grammar. The derivation tree consists of terminal and nonterminal symbols, where terminal symbols represent leaves of the tree. The derivation tree also contains helper nodes corresponding to EBNF features like repetition or optional elements.

One of the parameters that may be investigated, is a relative occurrence of symbols in derivation trees. Relative occurrence of a symbol in a program is defined as:

$$r_{sym} = \frac{n_{sym}}{N}$$

where n_{sym} means a number of occurrences of the sym symbol in the derivation tree and N represents a number of all symbols/nodes of the derivation tree.

Table I is the result of statistical analysis and represents 40 most frequent occurrences of particular symbols in all programs of our sample. As it can be expected, variable names and expressions have the greatest frequency. However, some symbols even did not occur in any of our sample programs, like `default`, `fbind`, `fpat` and `gdpat`.

It is possible to provide similar statistics for specially selected sample of programs within a specific domain. This might show, which language elements are used in programs of the domain and which elements can be omitted from the domain-specific dialect. Moreover, statistical analysis can also be used to partition a sample of programs into groups based on a usage of the language elements. The classification of programs depending on problems is just one of possible directions for the research in the future.

C. Pattern Recognition

To recognize syntactic patterns in a program or a set of programs, it is necessary to decide which parts of the programs may be considered similar. The simplest possibility is to consider only the equal trees, what is, however, exceedingly limiting. Trees can be considered similar if their structure is the same except for the attributes of terminal symbols (approach that was chosen).

Another approach is to allow differences in the whole subtrees rooted in a node of the same type. This would allow more complex syntactic variables, what is considered as harder to implement.

To find patterns in the program derivation tree, a simple algorithm can be used that is based on the function $findPatterns$ defined below:

$parents \leftarrow allParents(elements)$
$groups \leftarrow findGroups(parents)$
if $groups$ is empty **then**
  return $[groups]$
else
  for all $group \in groups$ **do**
    Add $findPatterns(group)$ to $foundGroups$
  end for
  return $mergeGroups(foundGroups)$
end if

Function $findPatterns$ takes a list of tree elements and recursively examines their parents to find a set of groups of subtrees having a similar structure. It uses helper functions with the following meaning:

- $allParents$ – returns a set of parents of all tree elements in a group;
- $findGroups$ – given a set of tree elements, returns list of groups of elements with similar subtrees;
- $mergeGroups$ – merges list of lists of groups into a single list.

To initiate the algorithm, the $findPatterns$ function is called on terminal symbols of the tree. Then it tries to walk up to the root of the tree while it can find groups of subtrees with similar structure.

As a result of the algorithm, a list of groups of subtrees is provided, where each group corresponds to a found pattern and contains all occurrences of the pattern.

D. Example

Lets look at a simple example program. It defines binary search tree and functions to manipulate with it. Derivation tree of this program is represented in Fig. 2.

```
data BStree a = Nil
    | Bin a (BStree a) (BStree a)

insert x Nil = Bin x Nil Nil
insert x (Bin y t1 t2)
    | x < y  = Bin y (insert x t1) t2
    | x == y = Bin y t1 t2
    | x > y  = Bin y t1 (insert x t2)

delete x Nil = Nil
delete x (Bin y t1 t2)
    | x < y  = Bin y (delete x t1) t2
    | x == y = join t1 t2
    | x > y  = Bin y t1 (delete x t2)
```

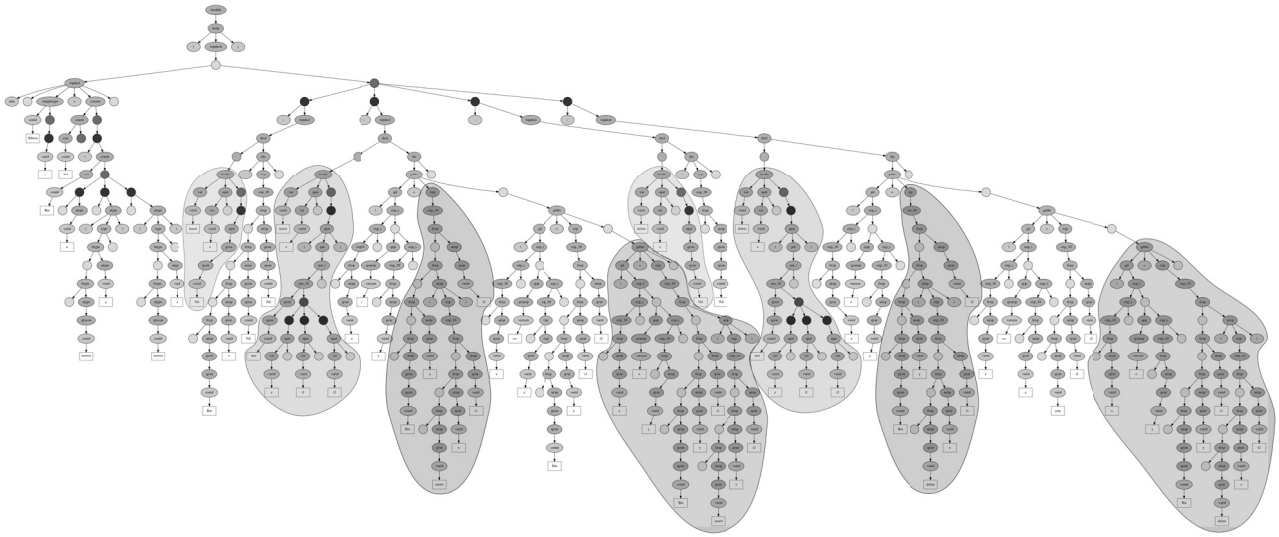


Fig. 2. Patterns found in derivation tree of the example program

Using the described method, it is possible to find several recurring patterns in this program (see Fig. 2). The most important are:

- `|` `x` α `y = Bin y t1 (` β `x t2 )`
- `Bin y (` α `x t1 ) t2`
- α `x ( Bin y t1 t2 )`
- α `x Nil`

Greek letters in the patterns regard the syntactic variables that can be replaced with concrete syntactic elements. Other identified patterns are too small to be mentioned. They are simply meaningless.

III. Language Inference

Grammar/language inference is a process of learning of a grammar/language from programs. It can be formulated as follows: Giving a set of positive samples $S^+ \subseteq L(G)$, $L(G) = \{ w \mid w \in L(G)\}$ and set of negative samples $S^- \subseteq \overline{L}(G)$, $\overline{L}(G) = \{ w \mid w \notin L(G)\}$, which might be also empty, find at least one grammar G such that $S^+ \subseteq L(G)$ and $S^- \subseteq \overline{L}(G)$.

Up to now, language inference has been used in several research domains such as language acquisition, pattern recognition, computational biology, and software engineering [6]. In language acquisition a child, being exposed only to positive samples, is able to discover the syntactic representation of the language (grammar). The aim of research on grammar inference is to provide different of models how language acquisition takes place. In pattern recognition, pattern grammars are used for pattern description and recognition. In computational biology grammar inference has been used for analysis of DNA, RNA, or protein sequences. For example, grammar inference has been successfully applied to predict secondary structures and functions of the biological molecules. An early application of grammar inference in software engineering was programming language design, where an inference algorithm for a very restricted grammar, operator precedence grammar, has been proposed.

A. MAGIc

MAGIc, **M**emetic **A**lgorithm for **G**rammatical **I**nferen**c**e, represents an memetic algorithm for grammar/language learning. Term memetic algorithm was first used in [14] based on word *meme* which was first coined in [15]. *Meme* represents a basic element of cultural evolution, similarly as *gene* represents a basic element of genetic evolution. Memetic algorithm in essence represents a search algorithm (mostly evolution based) with blended local search procedure. Other synonyms are hybrid evolutionary algorithms (EAs) or Lamarckian EA's. EAs were proven to be successful for many problem domains where other heuristic procedures fail. They give optimal or semi-optimal solutions. On other hand most EAs do not use all the problem knowledge as they could. Therefore hybrid algorithms were proposed with local problem specific subroutines, which give better solutions in less computational time.

Algorithm MAGIc (Fig. 3) is composed of next few steps:

- initialization,
- local search,
- mutation,
- generalization and
- selection.

In the initialization phase initial population is generated. On each true positive sample given as input the Sequitur [16] algorithm is used. Sequitur detects repetition of input symbols and factors it out by generating grammar rules/productions. Each initial grammar generated by Sequitur recognizes only one input sample from which it was generated. After the initial population is build, the evolutionary cycle takes place, where the local search, mutation, generalization and selection operators are used.

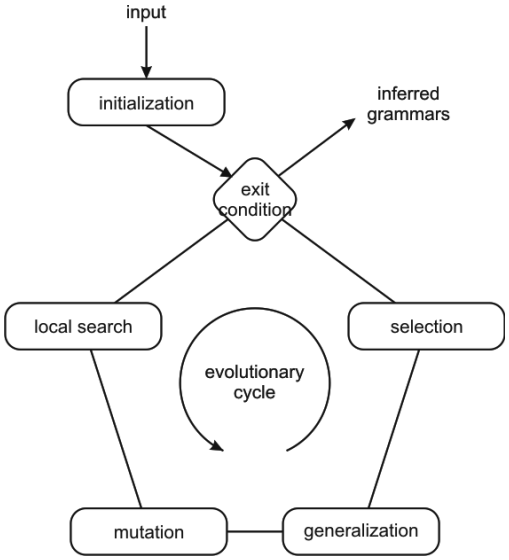


Fig. 3. Memetic algorithm for grammatical inference

The local search operator is the most important MAGIc step. It generates/produces new grammars that extends parsing capability to other true positive input samples that were previously discarded/not recognized by parent grammar. To change the grammar to parse more input samples the difference between true positive input samples and information about parse error position is used. To demonstrate the approach lets assume we have a grammar excerpt:

N_x `::= insert x (Bin y t1 t2)`

and samples:

```
insert x (Bin y t1 t2)
delete x (Bin y t1 t2)
```

Grammar rule/production N_x recognizes the first sample but fails to parse the second one. The difference between samples is in the first token `insert` $\leftrightarrow$ `delete`. By changing the grammar and adding new nonterminal N_y the new generated grammar can parse both input samples:

N_x `::=` N_y `x (Bin y t1 t2)`
N_y `::= insert`
N_y `::= delete`

Using this and similar grammar changing rule, the algorithm is able to produce grammars capable of parsing all true positive samples given as input. The interested reader is forwarded to [17] where all possible grammar changes are described in details.

The mutation operator was developed based on common properties used in programming languages, where some language construct can appear optionally, or can iterate. Therefore three different mutation solutions are proposed. Each grammar symbol (terminal or nonterminal) is randomly chosen for mutation. The selected symbol can be made optional, can appear zero or many times (iteration*) or one or many times (iteration$^+$).

MAGIc's generalization operator consists of few steps which are used to optimize and simplify generated grammars. In optimization phase the grammar is checked for recursion patterns and repeating patterns. The generated grammars are also simplified by removing equal grammar rules/productions and unit productions. Again, an interested reader is referred to [17], where generalization steps are described in detail.

In the selection step the best *N* grammars are selected into the next generation, where *N* represents population size given as algorithm input. Population individuals - grammars are sorted regarding their fitness value, that is calculated based on number of recognized/parsed input samples. MAGIc uses deterministic selection, which performed well, but still has some problems dealing with neutral solutions. The algorithm does not know how to differentiate between grammars that parse the same number of input samples, if their number is larger than new generation population size. In the near future we would like to experiment with other, more advanced selection techniques that would take in account the issue on neutral solutions as well.

B. Inferring data type declaration of Haskell language

MAGIc was already tested on different domain-specific languages (e.g. DESK, WHILE, HyperTree, FDL). The algorithm is able to infer context-free grammars from scratch [17] or can be used to extend existing grammars (given as algorithm input) [18]. This section presents results obtained by MAGIc for inference of part of Haskell language. Fig. 4 shows input samples of the algorithm. They represent data type declaration in Haskell language. Initial grammars were generated using input samples. The comparison of samples is done at the token level, not at the character level, therefore a lexical analysis phase is done beforehand. Regular expressions used in lexical analysis:

Token	data	id	int	string	=	|
Lexeme	data	[A-Z][a-z0-9]*	Int	String	=	|

1. `data MyType = Type`
2. `data Anniversary = Birthday | Wedding`
3. `data Review = Accept | Reject | Neutral`
4. `data ShapeType = Line Int | Point`
5. `data Component = Name String | None`
6. `data Temp = Celsius Int | Farenheit Int`
7. `data Paper = Custum Int Int | A4`

Fig. 4. Haskell data type declaration samples

Fig. 5 represents inferred grammar by MAGIc algorithm. Grammar was inferred in 6 generations from samples 7, 3, 4, 5, 1. Note, that not all samples were used in inference process. This is because of generalization step which can recognize repetition and recursion inside inferred grammars.

Control parameters of the algorithm were determined experimentally. They were: $population\_size = 70$,

```
N1 ::= data id = id N4
N2 ::= | id N2
N2 ::= | id
N2 ::= N3 N2
N2 ::= N3
N3 ::= int
N3 ::= string
N4 ::= N2
N4 ::= ε
```

Fig. 5. Inferred grammar

$mutation\_probability = 1\%$ and $max\_generations = 30$. On computer with *Intel Core 2 Duo* at $2.66GHz$ the average run on Haskell samples took about 8 seconds.

IV. Conclusion

The paper contributes to the abstraction phase in automatic roundtrip language engineering, which we can classify as unification of two phases: the phase of reverse abstraction - acquiring language patterns from an application samples and the phase of forward generation of new application from acquired language patterns.

While forward generation automated methods are well known, abstraction direction is more complicated, since it is not clear so far, if it can be fully automated.

In this paper we contribute to language abstraction showing two steps in abstraction automation - by deterministic analysis of program samples written in a given language as a first step, and grammatical language inference, as a second step. Moreover, we were focusing on Haskell language, and on grammatical inference on small part of Haskell language, data type declaration.

The future steps of our research will concentrate on extending the analysis to any language and on associating semantic rules when performing language inference.

Acknowledgment

This work was supported by SRDA Project of Slovak-Slovenian Research and Development Cooperation SK-SI-0003-10 and ARRS joint research project between Slovenia and Slovakia BI-SK/11-12-011: "Language Patterns in Domain-specific Language Evolution".

References

[1] F. P. Brooks, "No silver bullet: Essence and accidents of software engineering," *IEEE Computer*, vol. 20, no. 4, pp. 10–19, april 1987.

[2] P. Rechenberg, "Programming languages as thought models," *Structured Programming*, vol. 11, no. 3, pp. 105–115, 1990.

[3] M. P. Ward, "Language-oriented programming," *Software - Concepts and Tools*, vol. 15, no. 4, pp. 147–161, 1994.

[4] S. Dmitriev, "Language oriented programming: The next programming paradigm," Available: http://www.jetbrains.com/mps/docs/Language_Oriented_Programming.pdf, November 2004.

[5] P. Klint, R. Lämmel, and C. Verhoef, "Toward an engineering discipline for grammarware," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 14, no. 3, pp. 331–380, 2005.

[6] C. de la Higuera, *Grammar Inference: Learning Automata and Grammars*. New York, NY, USA: Cambridge University Press, 2010.

[7] S. Peyton Jones, *Haskell 98 Language and Libraries – The Revised Report*. Cambridge, England: Cambridge University Press, 2003.

[8] J. Ellson, E. Gansner, L. Koutsofios, S. North, and G. Woodhull, "Graphviz – open source graph drawing tools," in *Graph Drawing*, ser. Lecture Notes in Computer Science, P. Mutzel, M. Jünger, and S. Leipert, Eds. Springer Berlin / Heidelberg, 2002, vol. 2265, pp. 594–597.

[9] B. O'Sullivan, J. Goerzen, and D. Stewart, *Real World Haskell*, 1st ed. O'Reilly Media, Inc., 2008.

[10] P. Hercek, "Haskell 98 report," Available: http://www.hck.sk/users/peter/HaskellEx.htm, 2007.

[11] C. Donnelly and R. Stallman, *Bison: The Yacc-compatible Parser Generator*, 2010, available: http://www.gnu.org/software/bison/manual/.

[12] V. Paxson, W. Estes, and J. Millaway, *Lexical Analysis With Flex*, 2007, available: http://flex.sourceforge.net/manual/.

[13] S. Marlow, "The Haskell 2010 Language Report," Available: http://www.haskell.org/onlinereport/haskell2010/, 2010.

[14] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," California Institute of Technology, Concurrent Computation Program 158-79, Tech. Rep., 1989.

[15] R. Dawkins, *The Selfish Gene*. Oxford University Press, Oxford, UK, 1976.

[16] C. G. Nevill-Manning and I. H. Witten, "Identifying hierarchical structure in sequences: A linear-time algorithm," *Journal of Artificial Intelligence Research*, vol. 7, p. 67, 1997.

[17] M. Mernik, D. Hrnčič, B. R. Bryant, and F. Javed, "Applications of grammatical inference in software engineering: Domain specific language development," in *Mathematics, Computing, Language, and Life: Frontiers in Mathematical Languistics and Language Theory - Vol. 2, Scientific Applications of Language Methods*, C. Martín-Vide, Ed. Imperial College Press, 2010, pp. 421–457.

[18] D. Hrnčič, M. Mernik, and B. R. Bryant, "Embedding DSLs into GPLs: A grammatical inference approach," *Information Technology and Control*, to appear.

Form-Driven Application Generation: A Case Study

Sonja Ristić, Slavica Aleksić, Ivan Luković
University of Novi Sad
Faculty of Technical Sciences
Novi Sad, Serbia
(sdristic, slavica, ivan)@uns.ac.rs

Jelena Banović
Crnogorski telekom
Podgorica, Montenegro
jelenap@t-com.me

Abstract— **The paper presents a case study in the automated generating of software applications by using the IIS*Studio development environment. IIS*Studio is aimed to provide the information system design and generating executable application prototypes. Input data for our tool is the set of platform independent specifications. Among these specifications, a set of specified form types is particularly important. A form type is central IIS*Studio concept, used to model the structure and constraints of various business documents. Through the chain of transformations IIS*Studio generates a set of subschemas in the 3rd normal form and also a global relational database schema by integration of the subschemas. It enables a full implementation of database schemas under different target database management systems, by using its own SQL Generator. IIS*Studio also comprises a tool for the formal specification of business applications, and a generator of the executable application prototypes. The presented case study illustrates main features of IIS*Studio application generator tool, as well as the methodological aspects of its usage. The chain of transformations from a platform independent model, through the series of platform specific models with different degree of platform specificity, towards executable program code, is crucial in our approach, and the presented case study justify that claim.**

Form type, transformation chain, automated application generation, Model-driven Software Development (MDSD).

I. INTRODUCTION

Informally, a form is a way of organizing and presenting data. In the context of organization and its information system one can distinguish between business forms and computerized forms. Business forms (documents) are broadly used in organizations to conduct daily operations and to communicate with their affiliated entities (e.g. staff, superior managers, customers, suppliers, etc.). They may provide an important input source for database (db) schema design, since the most widely used data are gathered or reported in them ([10], [11], [12], [17]). There are tools (CASE or model-driven) aimed at automated application generation using set of forms as input source, such as *DeKlarit* [4]. On the other hand, users in organizations are experienced in handling and manipulation of databases through screen (computerized) forms which are the most natural interface between user and data. Database systems have been extended to deal with forms and other types of documents used to facilitate the manipulation of data via computerized form (e.g., Oracle SQL Forms, Informix-SQL, Microsoft Access Forms, different reports tools, etc.) [16].

Forms are objects, easy to read and understand, well structured and, consequently, easy to formalize. Therefore, business forms are a source for eliciting user information requirements and also for designing and developing user-oriented information systems. There are various research works about the use of forms (business and/or computerized) in different contexts. In order to integrate services in Office Information system, Tsichritzis in [21] introduces the concepts of form type, form template and form instance. In [19] and [20] Shu et al. proposed using forms to specify system requirements. Batini et al. in [6] and Choobineh et al. in [10] and [11] used business forms as input data for the process of database schema design based on generating entity-relationship (ER) diagrams. Choobineh and Venkatraman in [9] presented a methodology and tools for derivation of functional dependencies from business form. A form-based approach for reverse engineering of relational databases is proposed by Malki, Flory, and Rahmouni [16]. This methodology uses the information extracted from both form structure and instances as a database reverse engineering input using an interaction with a user. This approach inspired later research on extracting personalized ontology from data-intensive web application [7]. Namely, S. M. Bensliman, Malki, Rahmouni and Dj. Bensliman based their approach on the idea that semantics can be extracted by applying a reverse engineering technique on the structures and the instances of HTML-forms which are the most convenient interface to communicate with relational databases on the current data-intensive web application. This semantics is exploited to produce a personalized ontology. Tailoring some ideas from [22] and [23], Wu et al. in [24] presents a methodology that uses factoring and synthesis to process knowledge involved in forms for designing form-based decision support systems.

The objective of our research is to propose a form-driven approach to application generating. Through a number of research projects lasting for several years, we developed the IIS*Studio development environment (IIS*Studio DE, current version 7.1). It is aimed to provide the information system (IS) design and generating executable application prototypes. Input data for our tool is a set of platform independent specifications. Among these specifications, particularly important is the set of specified form types. A form type (Fig. 1, Fig. 2 and Section 3) is central IIS*Studio concept, used to model the structure and constraints of various business forms. Through a chain of transformations IIS*Studio generates a set of subschemas in the 3rd normal form and a global relational database schema by

Research presented in this paper was supported by Ministry of Science and Technological Development of Republic of Serbia, Grant III-44010, Title: *Intelligent Systems for Software Product Development and Business Support based on Models*.

integration of the subschemas. It also provides a full implementation of database schemas under different target database management systems, by using its own SQL Generator. IIS*Studio also comprises a tool for the formal specification of business applications, and a generator of the executable application prototypes.

The concept of a form type in our approach mainly corresponds to the concept of a business component that is the main modeling concept *DeKlarit* tool [4] relies on. *DeKlarit,* like the IIS*Studio, can generate relational database schema and SQL commands for various DBMSs. Unlike the *DeKlarit*, IIS*Studio further provides specific concepts and tools for the specification of the transaction programs and business applications ([14] and [18]). Our approach has some similarities with the approaches presented in [6], [8], [9], [10] and [11]. But, besides the set of functional dependencies F (like in [9]) the initial set of constraints, inferred from a form type, withal consists of: a set of non-functional dependencies NF, a set of special functional dependencies F_u, and a set of null value constraints N_c. Their detailed explanation with examples may be found in [12]. While in approaches presented in [6], [10] and [11] just ER diagrams are generated, IIS*Studio generates relational database schemas and executes an efficient transformation of design specifications into error free SQL specifications of relational database (db) schemas for different DBMSs ([1], [2] and [3]). Although our research does not tackle the same problems as it is in [7] and [16], a common value is a reported necessity of the integration of independently developed database subschemas. Integration of db subschemas in [7] and [16] is done at the conceptual level. In our approach it is done at the implementation instead of the conceptual level [15]. A db schema at the implementation level is expressed by the relational data model.

Our IIS*Studio comprises tools for: conceptual modeling of a database schema; automated design of relational database subschemas in the 3rd normal form; automated integration of relational database subschemas; automated detection of constraint collisions; generating XML specifications of an IS; full implementation of database schemas under different target DBMSs, by using its own SQL Generator; conceptual modeling of transaction programs, and business applications of an IS; user interface template specification; and generating functional application prototypes of an IS. In the paper the case study is presented that illustrates main features of IIS*Studio application prototypes generator tool, as well as the methodological aspects of its usage.

The paper is organized as follows. In Section 2 it is presented a real system of Safe House Center (SHC), whose information system is designed by IIS*Studio. Section 3 explains main concepts needed to understand generation of applications in IIS*Studio. Methodological aspects of the usage of IIS*Studio Application Generator are given in Section 4 through an example of *Donations* subsystem of the SHC information system. The last section concludes the paper.

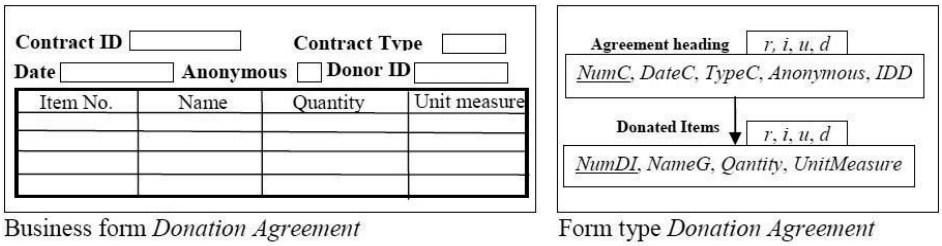


Figure 1. The business form *Donation Agreement* and its form type

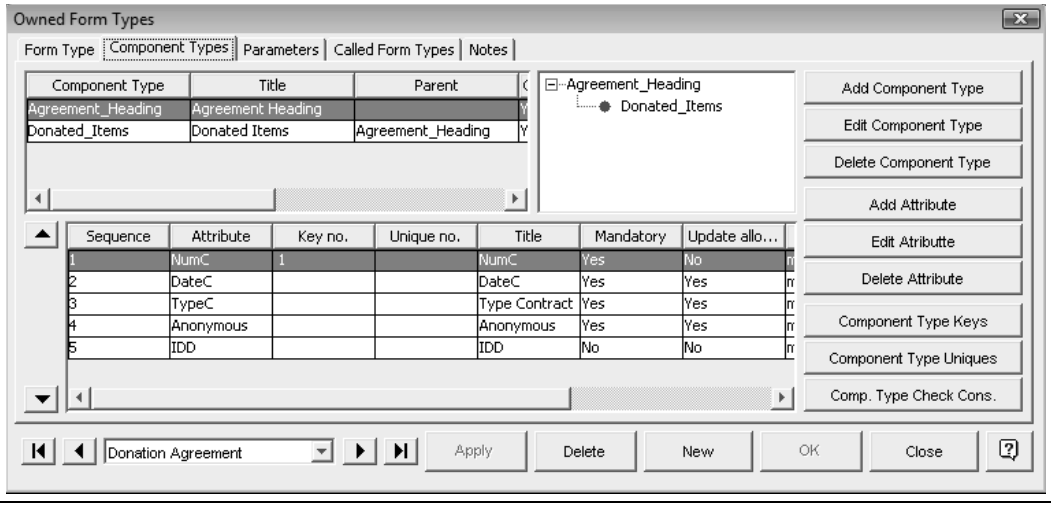


Figure 2. The IIS*Studio form for specification of the form type *Donation Agreement*

II. CASE STUDY: THE SAFE HOUSE CENTER

The Safe House Center (SHC) provides support for those children impacted by domestic violence. It offers a safe and nurturing environment where qualified staff assists youth with their basic needs, assessment, advocacy and stabilization. Besides, it organizes foster care and assists and supervises foster families. The SHC is on a great extent based on donations. In the paper we will present the *Donation* subsystem of the SHC information system. It is simple compared with the whole SHC information system, but suficiently complex to allow an ilustration of main features of IIS*Studio application generator tool, and the methodological aspects of its usage.

A donor may be a natural person, a legal entity or a group of citizens. A contribution to SHC may be made by donating: cash, check, goods, remedies, food, bequests, insurance, stocks, bonds or mutual funds, etc. All donations must be recorded. A donation agreement is made between a donor and SHC. In the agreement all donated items are specified. A donor may require staying anonymous, meaning that all external documents should not contain data about donor.

The business forms used in SHC important for *Donation* subsystem beyond others are: *Donor Card* (containing basic data about donor: name, type, contact data, etc.) and *Donation Agreement* (presented in Fig. 1). In the following sections we present the process of form-driven application generating by means of IIS*Studio and using the SHC case study.

III. IIS*STUDIO BASIC CONCEPTS

All the designers' specifications of an IS model created by IIS*Studio belong to an IIS*Studio **project**. Each project is organized as a tree structure of **application systems**, where each application system may contain an arbitrary number of form types (see Fig. 5).

A **form type** is the main modeling concept in IIS*Studio. Initially, each form type is an abstraction of a business form. However, it may be enriched by additional specifications that are not included in the entry business form, like specifications of: key and unique constraints; check constraints; allowed database CRUD (Create, Retrieve, Update and Delete) operations applied by means of screen computerized forms to manipulate data of an IS; functionalities concerning relationships between generated screen forms, i.e. transaction programs, etc. The business form *Donation Agreement* (*DA-bf*) is presented on the left-hand side of Fig. 1. It may be modeled by the form type *Donation Agreement* (*DA-ft*). The simplified representation of the structure of the *DA-ft*, which generalizes the *DA-bf*, is presented on the right-hand side of Fig. 1.

A form type is a hierarchical structure of form type components. Each component type is identified by its name within the scope of a form type, and has non-empty sets of attributes and keys, a possibly empty set of unique constraints, and a specification of the check constraint. A set of allowed database operations must be associated with each component type. If the *update* operation is associated with a component type, the set of updatable attributes of the component type must be specified as well. In addition, each attribute of a component type may be marked as mandatory or optional. The form type *Donation Agreement* (Fig. 1) has two component types: *Agreement Heading* and *Donated Items*. For both of them allowed operations are: read, insert, update and delete. An IIS*Studio form for the specification of component type *Agreement Heading* is presented in Fig. 2.

Each attribute of a component type is selected from a global set of all IS attributes. According to the universal relation scheme assumption present in the relational data model, the attributes are globally identified only by their names. IIS*Studio imposes strict rules for specifying attributes and their domains, and for specifying component type attributes.

In the traditional approaches to the IS design, database schema design not rarely precedes the specification of screen or report forms of transaction programs. On the contrary, in IIS*Studio a designer the first specifies screen and report forms, and indirectly, creates an initial set of attributes and constraints. The form type structuring rules provide automatic inference of relational db constraints from form types. These constraints are processed by the modified synthesis algorithm later in the process of a relational db schema design, as it is explained in details in [12]. The correspondence between a form type and a relation scheme is rarely one-to-one. In our simplified example it is one-to-two since the db schema generated just from the form type *Donation Agreement* contains two relation schemes: *Agreement_Heading* and *Donation_Item*. It is also possible for a relation scheme to contain attributes from different form types, making the aforementioned correspondence to be many-to-one [12]. By creating form types, a designer at the same time specifies: (i) a future database schema, (ii) functional properties of future transaction programs, (iii) and a look of the end-user interface.

A form type in IS design by means of IIS*Studio has a dual role. On the one hand it provides an important input data for database design, and on the other hand it is a source for the generation of a sole transaction program and its screen or report form. In order to enable formal specification of functionalities concerning relationships (so called "calls") between generated screen forms, i.e. transaction programs, a concept named **business application** (BA) is introduced.

The form for BA specification is given in Fig. 3. The rectangles in Fig. 3 represent form types, while the arrows represent calls between them. The form type *Catalog of Donations* is the entry point of business application since there is no arrow sinking in it. The form type *Donation Agreement* is called from the form type *Catalog of Donations* and the form type *Donator* and it may call the form type *Foster Family*. In the following section the further explanation of business application specification is given.

Scope of each business application created in IIS*Studio is an application system, because any business application belongs to exactly one application system. A business application specification is a source for generation of a program code that covers calls between generated transaction programs, i.e. their forms, and a synchronization of their behavior. The union of the sets of form types of a selected application system and all its application subsystems, alongside with the set of its business applications' specifications

represents a technology (platform) independent model (PIM) of the real system being observed.

IV. An Example of Application Generation by Means of IIS*Studio

IIS*Studio relies on the approach that conforms to the principles of model-driven approach. By means of IIS*Studio, a designer specifies only PIM models, because they are free of any implementation details. By the chain of consecutive transformations a set of different semi or fully platform specific models (PSMs) is generated ([12], [13], [14] and [15]). The chain of transformations in IIS*Studio can be divided into three subsets: (i) transformations aimed to generate a formal and an implementation DB schemas (DB schema generator); (ii) UI (User Interface) prototype generators; and (iii) application generators. The first and the third subset of transformations are mandatory in order to generate application prototype (see Fig. 4). The second subset is optional, and helps designer in the process of selecting the best fitting UI. Therefore, its presentation is omitted here. A case study illustrating the first subset of transformations may be found in [12] and [15]. In the following text a case study illustrating the third subset of transformations is presented.

The first step of the process of business application specification and generation by means of IIS*Studio is Project Specification (Fig. 4). Fig. 5 presents the *Safe House* project tree, containing one application system *Donation* with form types *Catalog of Donations*, *City*, *Donation Agreement*, *Donator* and *Foster Family* that are modeling corresponding business forms. The form types are specified through IIS*Studio screen forms, like the one in Fig. 2. After selecting the tab *Component type*, designer may insert/edit/delete component type, or/and insert/edit/delete the attributes (Fig. 6) of selected component type, or/and specify component type constraints (key, unique or check constraints).

During the process of attribute specifications one would specify the attribute title, behavior (is it modifiable or not), allowed operations, obligingness and default value through tab *Definition* (see the left-hand side of Fig. 6). This part of specification is important for the database schema generation process. The tab *Display* is important for the application generator, because it enables specifying the display characteristics of an attribute. The selected display option for the attribute *TypeD* is *ComboBox* (Fig. 6), and possible values are a natural person, a legal entity or a group of citizens.

The example of *Donation Agreement* form type illustrates the multiple role of form type. At the same time it is: a model of *Donation Agreement* business form (Fig. 1); a data model – one can see it as a conceptual database schema (Fig. 2); a model of computerized (screen) form (*Donation Agreement* screen form in Fig. 8); and a model of transaction program, since the functional properties of future transaction program are specified.

The form type structuring rules provide an automatic inference of the set of attributes and relational database constraints. A designer interactively may control the process of the relational database schema generation, giving the necessary additional information. This information is combined with the PIM (containing a set of fundamental concepts and a set of form types) that is automatically transformed into a relational database schema. A log file containing the records about the transformation process is generated, as well. The steps of a database schema design process in the IIS*Studio environment are presented and illustrated in detail in [12].

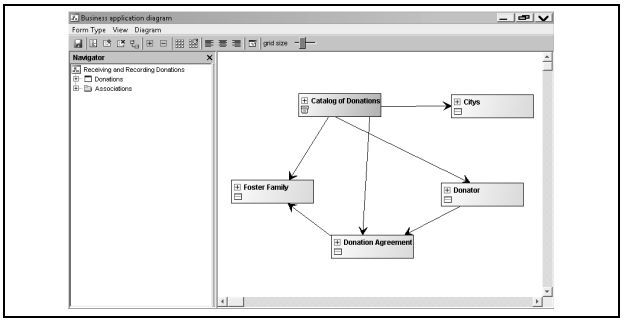


Figure 3. The IIS*Studio diagram of a business application *Donation*

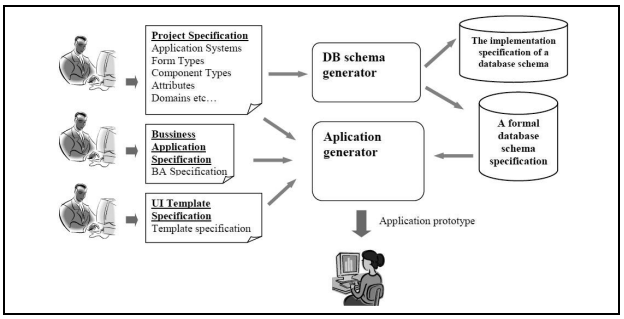


Figure 4. The chain of model transformations implemented in IIS*Studio

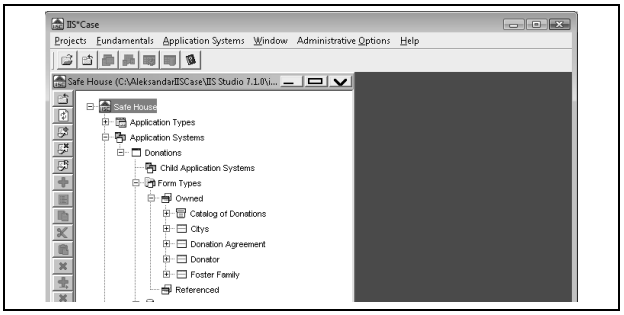


Figure 5. A segment of *Safe House* project tree

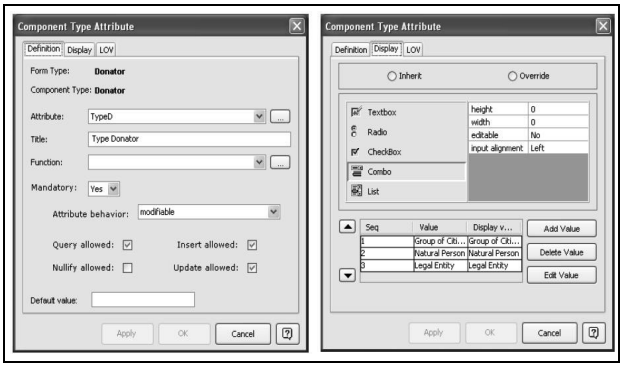


Figure 6. The attribute *TypeD* specification

The next step is business application specification. Specifying the form types must precede the design of a business application, because the specification of a business application in IIS*Studio comprises a structure of the selected form types. The screen form of the IIS*Studio tool for business application specification is presented in Fig. 3. The first step in specifying the business application structure is to select necessary form types. One of the selected form types has to be designated as an entry-point form type. The screen form generated from the entry-point form type is the first one accessible to end-users, when they initiate the business application, and by means of they can access the other (subordinated) forms in the application. In Fig. 3, the form type *Catalog of Donations* is declared as the entry-point. This form type is specific since it is menu form type. The application generator will transform it into menu screen, aimed at selecting possible further choices (see Fig. 8).

After the selection of form types, it is necessary to specify their mutual relationships ("calls") so as to specify the business application structure. In the business application diagram in Fig. 3 the arrow between the *Donator* and *Donator Agreement* form types indicates a call specification within a business application. In this case, *Donator* is the *calling form type*, and *Donator Agreement* is the *called form type*. By selecting a form type and *Edit* option the form types that are to be called from a selected form type are specified (Fig. 7). The consequence of such a specification is that the screen form generated from the calling form type has to support calls of the screen form generated from the called form type. For example, clicking the button *Donation Agreement* on the screen form *Donator* causes calling the corresponding screen form (Fig. 8). Besides, each call specification in IIS*Studio has the following properties (Fig. 7): *Passed values* – the list of passing values from the calling to the called form types; *Calling mode* – the rules for data selection and transferring data from the calling to the called form type; *Calling method* – a behavior of the calling and the called form type; and *UI positioning* – positioning properties of the UI control item for executing the call.

A business application specification can be regarded as the dynamic part of a GUI (Graphic User Interface). In order to generate appropriate transaction programs, the static aspects of GUI may be specified as well. All visual (displayable) elements are a part of GUI static aspects. The IIS*UIModeler is an integrated part of the IIS*Studio DE, aimed at modelling of GUI static aspects. By means of IIS*UIModeler a designer specifies UI templates. UI template specification contains attribute values that describe common UI characteristics, such as: screen size, main application window position, background/foreground colour, etc. The set of UI template attributes can be separated into six groups: global attributes, screen form attributes, table attributes, panel attributes, display/update attributes and button attributes. The specification of the UI template is stored in the IIS*Studio repository.

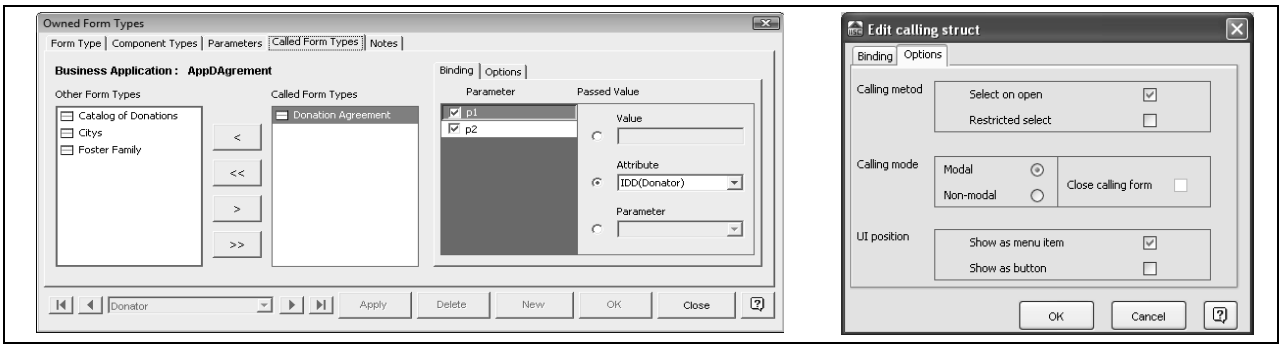


Figure 7. IIS*Studio form for call specification

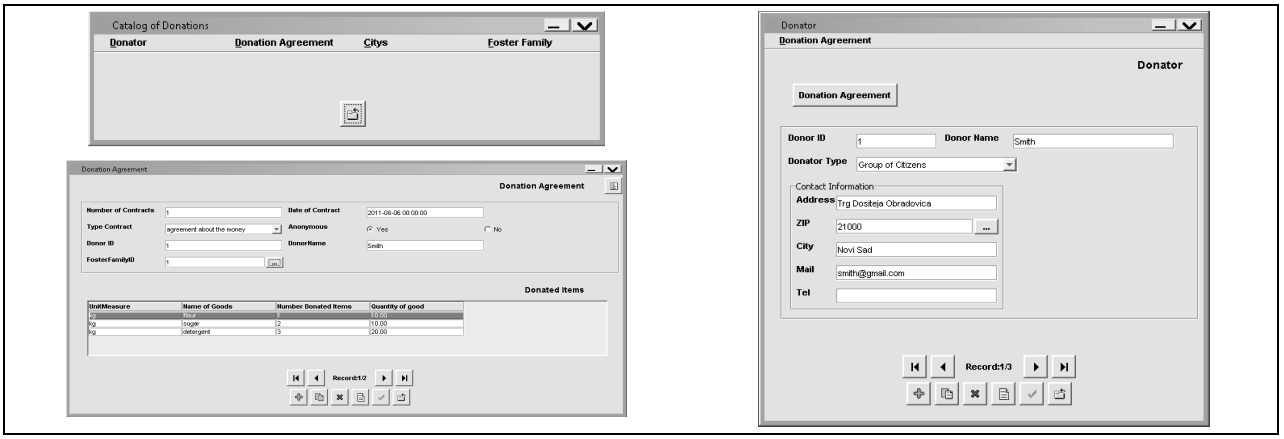


Figure 8. The screen forms of *Donation* application

UI template specifications are independent from any specific IS project specification, generated by means of IIS*Studio tool. The same UI template may be used for application prototype generation of different ISs, as well as the same IS project specification may be transformed in different ways by means of different UI templates. The specification of UI template may be seen as a fully platform independent UI model. The detailed description of the IIS*UIModeler and UI template attributes may be found in [5].

A business application specification together with the specifications of selected UI templates is a source for the generation of a program code that covers calls between generated transaction programs, i.e. their forms, and a synchronization of their behavior. The process of transaction program generating has three steps: (i) generating of the subshemas of the form types incorporated in a business application; (ii) generating of the UIML (User Interface Markup Language) application prototype specification alongside with generating of the executable user interface Java code from the UIML specification; and (iii) generating of the executable business application specification. Some of the generated screen forms of *Donation* application that supports *Donation* subsystem of *Safe House* IS are presented in Fig. 8.

V. CONCLUSION

The form-based approaches in the software engineering are present for more than two decades. Some of them are for: database analysis and design; extracting object-oriented db schemas from relational databases; extracting personalised ontology from data-intensive web applications; designing form-based decision support systems; etc. In our approach we are using the notion *form-driven*, instead of form-based, inspired with the different meaning of the model-based and model-driven approaches. A form type is the central concept of our IIS*Studio tool aimed at automated IS design and application generation. By creating form types, a designer specifies at the same time: (i) a future database schema, (ii) functional properties of future transaction programs, and (iii) a look of the end-user interface. One of the main assumptions of the model-driven approach to software system development is that software systems of large complexity can only be designed and maintained if the level of abstraction is considerably higher than that of programming languages. By means of models, semantics in an application domain can be precisely specified using terms and concepts the end-users are familiar with, such as the business forms. The focus of software development is shifted from the technology domain toward the problem domain. In our future work, in order to justify the correctness of the *form-driven* adjective we aim to investigate the reverse influence of the changes in the database schema to the screen and business forms.

REFERENCES

[1] S. Aleksić, I. Luković, P. Mogin, and M. Govedarica, "A Generator of SQL Schema Specifications," Computer Science and Information Systems (ComSIS), Vol. 4, No. 2, 2007, pp. 77-96.

[2] S. Aleksić and I. Luković, "Generating SQL Specifications of a Database Schema for Different DBMSs", *Info M-Journal of Information Technology and Multimedia Systems*, No. 23, 2007, pp. 36-43.

[3] S. Aleksić, S. Ristić, I. Luković, "An Approach to Generating Server Implementation of the Inverse Referential Integrity Constraints", The 5th International Conference on Information Technologies ICIT 2011, May 11th – 13th, Amman, Jordan, 2011, Proceedings on CD.

[4] ARTech. *DeKlarit*™, Chicago, U.S.A. [Online]. May 2010, Available: http://www.deklarit.com/

[5] Banovic J, "An approach to Generating Executable Software Specifications of an Information System", Ph.D. Dissertation, University of Novi Sad, Faculty of Technical Science, Serbia, 2010.

[6] C. Batini, B. Demo, A. Di Leva, A methodology for conceptual design of office data bases, Information Systems 9 (3/4), 1984, pp. 251– 263.

[7] S. M. Benslimane, M. Malki, M. K. Rahmouni, and Dj. Benslimane, Extracting Personalised Ontology from Data-Intensive Web Application: an HTML Forms-Based Reverse Engineering Approach, INFORMATICA, Vol. 18, No. 4, 2007, pp. 511–534.

[8] T. Catarci, M.F. Costabile, S. Levialdi, and C. Batini, Visual query systems for databases: a survey, Journal of Visual Languages and Computing 8, 1997, pp. 215– 260.

[9] J. Choobineh, and S.S. Venkatraman, A methodology and tools for derivation of functional dependencies from business form, Information Systems 17 (3), 1992, pp. 269–282.

[10] J. Choobineh, M.V. Mannino, J.F. Nunamaker, and B.R. Konsynski, An expert database design system based on analysis of forms, IEEE Transactions on Software Engineering 14 (2), 1988, pp. 242–253.

[11] J. Choobineh, M.V. Mannino, and V.P. Tseng, A form-based approach for database analysis and design, Communications of the ACM 35 (2), February 1992, pp. 108–120.

[12] I. Luković, P. Mogin, J. Pavićević, and S. Ristić, "An Approach to Developing Complex Database Schemas Using Form Types", Software: Practice and Experience, John Wiley & Sons Inc, Hoboken, USA, DOI: 10.1002/spe.820 Vol. 37, No. 15, 2007,pp. 1621-1656.

[13] Lukovic I., Popovic A., Mostic J., Ristic S. "A Tool for Modeling Form Type Check Constraints and Complex Functionalities of Business Applications", Computer Science and Information Systems, Special issue Vol 7. No. 2 "Advances in Languages, Related Technologies and Applications", DOI: 10.2298/CSIS1002359L, May, 2010, pp. 359 − 385.

[14] I. Luković, S. Ristić, S. Aleksić, J. Banović, A. Popović, "A Chain of Model Transformations in IIS*Case", Scripta Scientiarum Naturalium, University of Montenegro, Vol. 1, No. 1, 2010, pp. 59 − 76.

[15] I. Luković, S. Ristić, P. Mogin, and J. Pavicević, "Database Schema Integration Process – A Methodology and Aspects of Its Applying," *Novi Sad Journal of Mathematics*, Faculty of Science, Novi Sad, Serbia, ISSN: 1450-5444, Vol. 36, No. 1, 2006, pp. 115-140.

[16] M. Malki, A. Flory, and M. K. Rahmouni, Extraction of Object-oriented Schemas from Existing Relational Databases: a Form-driven Approach, INFORMATICA, Vol. 13, No. 1, 2002, pp. 47–72.

[17] P. Mogin, I. Luković, Ž. Karadžić,"Relational Database Schema Design and Application Generating Using IIS*CASE Tool", Proceedings of International Conference on Technical Informatics, Timisoara, Romania, 16-19. 11. 1994, Vol. 5, pp. 49-58.

[18] A. Popovic, "A Specification of Visual Attributes and Structures of Business Applications in the IIS*Case Tool,", M.Sc. (Mr) Thesis, University of Novi Sad, Faculty of Technical Sciences, 2008.

[19] N.C. Shu, FORMAL: a form-oriented, visual-directed application development system, Computer, 1985, pp. 38– 49.

[20] N.C. Shu, V.Y. Lum, F.C. Tung, and C.L. Chang, Specification of forms processing and business procedures for office automation, IEEE Transactions on Software Engineering SE-8 (5), 1982, pp. 499–512.

[21] D. Tsichritzis, Form management, Communications of the ACM 25 (5), July 1982, pp. 453–478.

[22] J.H. Wu, SDSS basis and application—a case study of the Taiwan Provincial Government, Journal of Chinese Institute of Industrial Engineering 13 (3), 1996, pp. 203– 213.

[23] J.H. Wu, A visual approach to end user form management, Journal of Computer Information Systems 41 (1), Fall 2000, pp. 31– 39.

[24] J. H. Wu, H. S. Doonga, C. C. Leeb, T. C. Hsiac, and T. P. Liang, A methodology for designing form-based decision support systems, Decision Support Systems 36, 2004, pp. 313–335.

Ensuring The Data Integrity And Credibility Based On Encryption And Decryption Algorithms

Liberios Vokorokos
Department of Computer and Informatics
Faculty of Electrical Engineering and Informatics
Košice, Slovak Republic
liberios.vokorokos@tuke.sk

Peter Fanfara
Department of Computer and Informatics
Faculty of Electrical Engineering and Informatics
Košice, Slovak Republic
peter.fanfara@tuke.sk

Eva Danková
Department of Computer and Informatics
Faculty of Electrical Engineering and Informatics
Košice, Slovak Republic
eva.dankova@tuke.sk

Branislav Madoš
Department of Computer and Informatics
Faculty of Electrical Engineering and Informatics
Košice, Slovak Republic
branislav.mados@tuke.sk

***Abstract*—Data security is a section of informatics, which is in progress and is still being developed. Increase of computing power enhances the risk of attacks for obtaining data of any type. Paper attention in mainly devoted to data confidentiality, integrity and sender authentication based on asymmetric encryption with using hash function and generators of very large random prime numbers to calculate public and private keys.**

Keywords-data integrity and confidentiality; hash function; random prime number generator; digital signature; asymmetric encryption algorithm;

I. INTRODUCTION

Nowadays there are many ways how to protect privacy. One possibility is using cryptography. Miscellaneous secrets, classified documents and reports, which shouldn't get into the hands of unauthorized persons at any price, have given suggestions to the emergence of confidentiality and encryption. Cryptography provides a solution to implement these services, which are based on knowledge of number combinations, called keys, which are unknown and heavy to guess for the attacker. The method of generation and distribution began intensively explored with the development of cryptography and its wider usage. Cryptography has become a modern science with practical implications for privacy and for increasing the security of electronic assemblies and security of electronic communications.

Modern cryptography is associated with the development of electronic forms of communication, where the interception of transmitted messages is relatively simple and technically feasible. The electronic form of communication that has gone from wired to wireless requires improvements in encryption algorithms. Transmission channel is easy to eavesdrop and therefore it is practically impossible to protect transmitted message. Modern forms of communication gave rise to public key cryptography, which solved the problem of key distribution over a public channel and enable communication with an unlimited number of participants. [1]

Following chapters describe the function and principles of asymmetric encryption algorithms, hash functions, generators of prime numbers and application for messages authentication.

II. ASYMMETRIC ENCRYPTION AND ALGORITHMS

Public key algorithms use two different but together related keys for the encryption and decryption. Encryption key (public key) and encryption algorithm are known. Decryption key (private key) knows only its owner (Figure 1). Public and private key are tied together by a strong one way function. It is computationally impossible (in a reasonable time) to derive the private key from the known public key. [2]

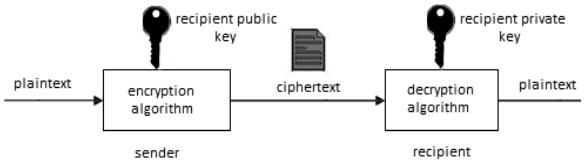


Figure 1 Principle of asymmetric encryption

Public key cryptography provides encryption and authentication functions, both functions are based on the fact that each participant of communication owns private and public key. Authentication in cryptography with public key encryption runs inversely against the encryption process (Figure 2). If the sender encrypts the plaintext with his private key, ordinary decryption can be realized only by the sender's public key (excluding few algorithms, e.g. RSA), which means that the sender truly encrypted message. In this case, the encrypted text (ciphertext) has the character of the digital signature and at the same time, it follows that the modification of the message (plaintext) without access to the private key is impossible therefore this cryptosystem with public key

provides message authentication and integrity of transmitted data. [2]

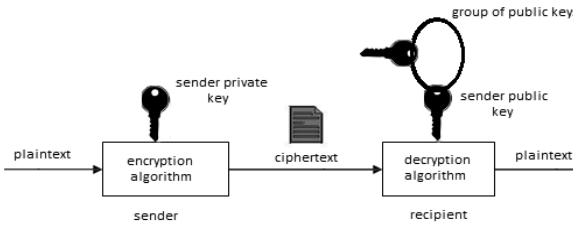


Figure 2 Authentication of sender via digital signature

Public-key algorithms can be divided into three basic groups: [1]

- Algorithms based on the problem of factoring numbers (e.g. RSA) – for the number to find its decomposition into primes.
- Algorithms based on the discrete logarithm problem (e.g., Diffie-Hellman, DSA) – for the numbers a, b and c to find the exponent b, if given a and $c=a^b$ in arbitrary cyclic group.
- Algorithms based on elliptic curves (II.A) encryption systems with elliptic curves are the latest group practically used in public key algorithms that have received recognition, including standardization.

In all algorithms are performed complex operations with very large numbers. For algorithms with numbers factorization and discrete logarithm is operand length in the range 512 - 4096 bits. The algorithms based on elliptic curve have similar operand length within 160 - 512 bits. Public key systems therefore have a major disadvantage because they are computationally very intensive and slow. [3]

A. Algorithm Based On Eliptic Curves

Elliptic Curve Cryptography (ECC) is a new and perspective way in modern cryptography. The main advantage of the algorithm is the fact there is no known algorithm that is able to solve the discrete logarithm in subexponencional time in the multiplicative group of finite fields. [4]

Major advantage in comparison with existing cryptographic systems and algorithms (e.g. RSA) is that ECC allows achieving the same cryptographic security with smaller key length. Some basic cryptographic algorithms can be easily modified for use in a system with ECC. A crypto system based on ECC has become a part of many cryptographic standards and represent an alternative to systems based on RSA and DSA algorithms. The ECC main advantages are higher speed and lower hardware requirements. [4]

Although known for several attacks on the discrete logarithm solution in the system with elliptic curves. if we use sufficiently large prime number neither of them cannot be used practically. Despite the low level of reconnaissance ECC while preventing wider deployment of systems with ECC. [3]

III. HASH FUNCTIONS

Cryptographic hash functions are a specific group of cryptographic algorithms. In mathematical terms this is a one-way function of the input data of different lengths extracted information with a fixed length. [6]

The original information cannot be entire contained in the hash code, so it is impossible to obtain the original message from the hash code. Hashed value is only a message digest of a message. For human, the hash value may correspond to a fingerprint. It is important to note that from the fingerprint cannot be made a human, but it can be clearly identified. Change any bit, respectively bits of the message will change the entire hash code, which has the characteristics of cryptographic checksum, making it suitable for ensuring the integrity of the message. [6]

Authentication of messages with symmetric encryption is based on the fact that only entities A and B possess a secret key. So the message originates from A and has not been modified, if the test of identity entire hash code of the receiving side is positive. Applying encryption to the entire message and the hash code guarantees the confidentiality of message. In asymmetric encryption to encrypt the entire hash code is used the sender's private key. This procedure provides a digital signature, since only the sender can create an encrypted hash code. [9]

Cryptographically strong hash function h must fulfill: [9]

- for any message m can be efficiently calculated hash value $h(m)$,
- for any hash value y is a difficult for computational resources within a reasonable time to find a message m, which is valid for $y = h(m)$,
- it is computationally difficult to find two different messages $m1 \neq m2$ such that true $h(m1) = h(m2)$.

By selecting a particular hash function defines a particular instance hash technique. General hash model is shown in (Figure 3). [9]

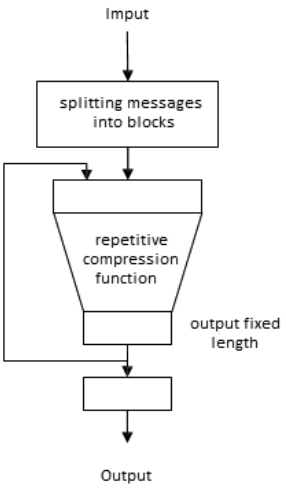


Figure 3 A general model of iterative hash function

Other well-known hash functions include Message-Digest algorithm 5 (MD5). Its use in applications where is required collision resistance is no longer recommended, however it still can be used, e.g. for extracting randomness in pseudorandom generators (IV.A). [9]

A. SHA Hash Function Group

Secure Hash Algorithm (SHA) was established and published by National Institute of Standards and Technology (NIST) standard specification FIPS-180, the revised version was published under the name FIPS 180-1 and is referred to as SHA-1. SHA-1 algorithm allows processing the original message with a maximum length of less than 2^{64} bits and constitutes output hash code with a length of 160 bits. The input message is processed by blocks of size 512 bits. [9]

After upgrading FIPS 180-1 to FIPS180-2 have been introduced three new hash algorithms SHA-256 (imprint length 256 bits), SHA-384 (384 bits) and SHA-512 (512 bits). The most significant difference in these algorithms is the length of the entire hash code algorithm that determines the security against brute-force method of attack, while the structure of algorithms is almost the same (TABLE I). [6]

B. Attacks On Hash Function

There are two types of brute-force attacks on hash functions. In the first case the attacker tries to create a message *m2* similar to message *m1*, to which it applies ($h(m1) = h(m2)$). The second case, when an attacker is able to find two arbitrary different sets of data ($m1 \neq m2$) for which the output from the same hash function will be ($h(m1) = h(m2)$) is called a collision and it is much simpler than the first type of attack. [9]

TABLE I Parameters of hash functions [6]

	SHA-1	**SHA-256**	**SHA-384**	**SHA-512**
Hash code length	160	256	384	512
Message length	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block size	512	512	1024	1024
Word size	32	32	64	64
Equivalent security in bits	80	128	192	256

IV. Generator of Random Numbers

Cryptographically secure random number generators are an important part of cryptographic systems. Random numbers are required for key generation and establishing communication between various participants.

Generating random numbers has character of generating sequence of random numbers which must fulfill two basic requirements of randomness and unpredictability. [5]

Although there are many protocols and encryption algorithms that use random numbers, the way of their generation is not yet determined by any standard. The more important question is therefore becoming - test of randomness generators and how they are implemented on different platforms. Generators can be divided into: [5]

- True Random Number Generators (TRNGs) – sometimes called physic or nondeterministic.
- Pseudorandom Number Generators (PRNGs) – also called deterministic.
- Hybrid generators.

A. Deterministic Generators

PRNG is an algorithm that can create from a short input sequence of random bits a very long sequence, which appears to be random. PRNGs are, despite for a lower price of security level, replacing in digital systems difficult to implement and too slow TRNGs. The most commonly used are PRNGs using e.g. hash functions (III), cellular automata, quadratic residue generators, etc. [8]

PRNGs advantage is the simple implementation in software or digital circuits, which themselves are deterministic systems. Current PRNGs generating output sequences with very good statistical characteristics and are much faster compared to TRNGs. [8]

B. Hybrid Generators

Represent a combination of physical and deterministic generator, as well as combining the advantageous properties (speed and quality of statistical properties of deterministic generator with nondeterministic output of the physical generator). Hybrid generators are justified in cases where we require random numbers at a faster speed than is capable of generating a physically generator. In this case, the output of the physical generator used for regular updating of the internal state of a deterministic generator, thus ensuring an increase in entropy output values. [13]

V. Ensuring The Data Integrity And Credibility

When generating a digital signature for an input parameters are used signed document and private key for asymmetric encryption algorithm. One of several requirements for the private key is to be stored safely. Whoever will receive private key, he would be able to generate a valid signature and could be misused the private key for signing other documents. Therefore, it is questionable whether it is appropriate to store the private key and enter it explicitly in the application that generates a digital signature. A cryptographic algorithm based on public key requires the selection of one or more large primes, and this choice should be random.

Strategy of generating large prime numbers is based on the fact that randomly generated large number is tested for primality. [10] All probabilistic prime tests determine primality only with a certain probability. There are deterministic prime tests too, which are much slower (e.g.

correct results gives the sieve of Eratosthenes, but there are faster deterministic tests like deterministic prime test with polynomial running time). However it is still rather slow nowadays, much slower than the probabilistic tests. Algorithms differ in the percentage of success in finding primality or time-consuming given locate primes. [11]

Miller-Rabin algorithm is one of the algorithms with the highest percentage accuracy of determining the random odd number is actually prime number. This causes the input condition, which differs from Fermat's algorithm, by the congruence is equal to the number *1* and *(n-1)*, where *n* is an input parameter. There is a possibility to enter a security parameter that determines the number of cycles for a random number which is used as input in the congruence input factor. Another advantage is that, for any random composite number *n* is the probability that the Miller-Rabin test *n* number identified as the prime, less than $(1/4)^{\text{security parameter}}$. [5]

As the encryption algorithm should be chosen Elliptic Curve Digital Signature Algorithm (ECDSA). This algorithm is the most advantageous of all algorithms. Its security is based on Elliptic Curve Discrete Logarithm Problem (ECDLP). [12] The complexity of solving the problem of discrete logarithms in elliptic curves is exponential. It has one disadvantage - it is not yet sufficiently explored from a mathematical point of view. On the other hand, advantage is using a shorter key (160 to 512 bits) against RSA and has less time consuming by calculation. Key length of at least 160 bits in the ECC key length is equivalent to 1024 bits in RSA with the same level of security, resulting in higher speed and lower hardware requirements.

Algorithms for generating prime numbers, keys and creating signature work with large numbers (several 100 series). Representing this numbers with the standard data types is impossible. The solution is a representation of an integer or a character array, each element of the array represents a single digit, e.g. number A = 123456789123456789 can represent by the field as follows:

$$A [ ] = \{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$$

Representation of numbers is in the decimal form. After that for these numbers must be defined basic mathematical operations: addition, subtraction, multiplication, division (modulo), congruence, copy (these operations are no longer trivial).

To generate the key is created the first application in which primes are generated using the modified Miller-Rabin algorithm that randomly generates two 100 local decimal numbers *p*, *q*. Generate a random initialization of input initialization vector (seed) from the system clock for the pseudo-random generator. Primes are used for calculating their multiplication $n = p*q$ (length *n* can be up to 200 numbers - about 200/log 2 = 664 bits - length of the key encryption algorithm). Key size is suitable to use 512 bit hash function.

Exponent *e* is generated for the public key (*e* should not be a prime), which may be small value (e.g. 3, 7, 11, 13, etc.) and condition, *e* is no dividing by value *(p-1)*(q-1)*, must be valid. As the next step will be calculated private key *d* from the equation $e*d \equiv 1 \pmod{(p-1)*(q-1)}$ using Euclid's algorithm. Generated keys are stored in the header file in the form of initialized global variables and arrays. The generated header file is then used to join the compilation process and the values to create a second application that is used to generate the signature. All keys are therefore already part of the application for generating signature (included in .exe file to be invisible). Hereby is ensured the implicit entering keys. Application of signature generating only loads the file with a document to be signed, calculates hash code file using a set of hash functions SHA-512 and encrypts the hashed code with encryption algorithm (e.g. RSA) using its private key. Encrypted hashed value file is then set as its digital signature that is stored in the generated file.

When generating a primes and subsequent determination of the key, key generation application generates an 8-digit sequence called PIN code, which is also enrolled in the header file and also enters into the compilation process for generating digital signature application. PIN code is generated randomly and without it is impossible to generate the signature - this feature will ensure the application against the abuse at the signing of other documents by unauthorized persons.

Verification can run via third set of applications allowing loading of the document and signature file. Process of verifying the digital signature will run using the explicitly specified public key. It is a condition that must be used the same algorithms as applied to generate a digital signature (encryption algorithm and hash function). The application consists of calculating hashing value loaded document, decryption algorithm with loaded signature and using the public key. If the resulting test of positive identity is true, then the signature is verified.

Principles of key generation application, digital signature application and links between them are shown in (Figure 4).

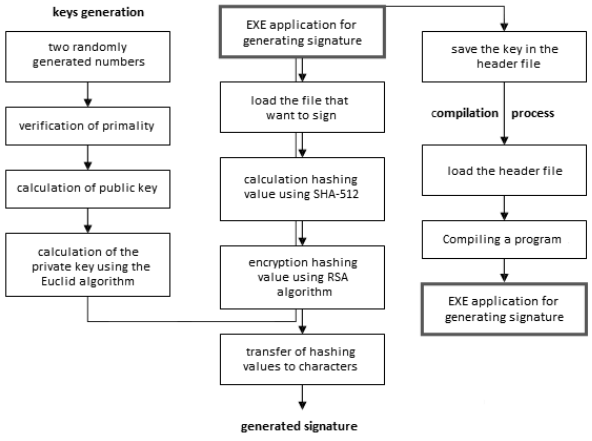


Figure 4 The principle of application for key generation and generation of electronic signature

Process of signature verification is shown in (Figure 5).

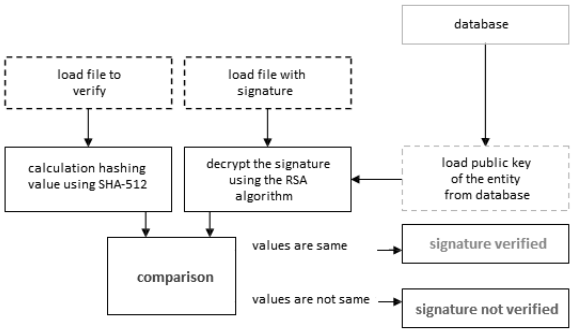


Figure 5 The principle of application for the signature

VI. CONCLUSION

The safety of all system with any cryptographic public key depends on several factors. These factors include particularly indisputability, mathematical algorithm used, the secure management of cryptographic keys and security system implementation in a particular application. The parameters of the current asymmetric encryption algorithms are designed that the increasing of computing power cannot be a threat to digital signature security consideration in a reasonable timeframe (tens of years). That follows from the fact that an attacker who does not know the private key, cannot calculate a digital signature of that entity, as well as true the fact that it is computationally impossible (within a reasonable time) to derive the private key from the public key. This means that the digital signature cannot be a threat to be forged. Nor is it possible to modify the signed document so that the same signature to it was consistent. Any interference with the document will undermine the integrity, the output will be different, hash value and signature will not be successfully verified. This document will be automatically regarded as irrelevant. Verifying the digital signature also ensures data credibility and authentication of the sender.

Results obtained in the generation of prime numbers (which calculated keys) and generating the signature are in terms of time processing acceptable. Time-consuming in this case depends on the performance of hardware.

Using digital signature is nowadays broad spectrum and flexible of reasons constantly evolving user requirements and the required security. Due to the nature of digital signatures, privacy issues and information security, however, remains despite the use of cryptography still open. Together with the development of procedures to ensure safety procedures are developed processes specialized for its weakening or completely eliminating.

ACKNOWLEDGMENT

This research has been supported by the Scientific Grant Agency of Slovak Republic under project VEGA No.1/0026/10 *"Modeling and simulation of security attacks in distributed computing environments and networks"* and this work was also supported by the Slovak Research and Development Agency under contract No. APVV-0008-10.

REFERENCES

[1] D. Levický, "Kryptografia v Informačnej bezpečnosti," Vol. 1, Elfa s.r.o., Košice 2005, ISBN 80-8086-022-X.

[2] Ch. Paar, J. Pelzl, "Introduction to Public-Key Cryptography", chapter 6 of "Understanding Cryptography, A Textbook for Students and Practitioners", [Online], Springer 2009, Available: http://crypto.rub.de/Buch/movies.php

[3] I. Blake, G. Seroussi, N. Smart, "Elliptic Curves in Cryptography", LMS Lecture Notes. Cambridge University Press, 2000, ISBN 0-521-65374-6.

[4] R. Crandall, C. Pomerance, "Chapter 7: Elliptic Curve Arithmetic". Prime Numbers" A computational Perspective (1st ed.), Springer-Verlag, pp. 285-352, ISBN 0-387-94777-9.

[5] P. Vondruška, "Crypto-world.info" [Online], IX. ročník, ISSN 1801-2140. Available: http://crypto-world.info.

[6] National Institute on Standards and Technology Computer Security Resource Center, [Online], NIST's Policy On Hash Functions, accessed March 29 2009, Available: http://csrc.nist.gov/groups/ST/hash/policy.html.

[7] M. Locktyukhin, K. Farrel, "Improving the Perfomance of the Secure Hash Algorithm (SHA-1)", [Online], Intel Software Knowledge Base (Intel), retrieved 2010-04-02, Available: http://software.intel.com/en-us/articles/improving-the-performance-of-the-secure-hash-algorithm-1/.

[8] J. Viega, "Practical Random Number Generation in Software", [Online], in Proc. 19th Annual Computer Security Application Conference, Dec. 2003, Available: http://www.acsac.org/2003/papers/79.pdf.

[9] I. Mironov, "Hash functions: Theory, attacks, and applications", [Online], Microsoft Research, Silicon Valley Campus, November 2005, Available: http://research.microsoft.com/pubs/64588/hash_survey.pdf.

[10] B. Green, T. Tao, "The primes contain arbitrarily long arithmetic progressions", [Online], Annals of Mathematics, 2008, p.481-547, Available: http://arxiv.org/abs/math.NT/0404188.

[11] O'Neill, E. Melissa, "The Genuine Sieve of Eratosthenes", [Online], Harvey mudd College, Claremont, CA, U.S.A, November 2008, Available: http://www.cs.hmc.edu/~oneill/papers/Sieve-JFP.pdf.

[12] F. Vercauteren, "Elliptic Curve Discrete Logarithm Problem", [Online], Kotholieke Universiteit Leuven, 2005, Available: http://homes.esat.kuleuven.be/~fvercaut/talks/ECDL.pdf.

[13] T. Saito, K. Ishii, I. Tatsuno, S. Sukagawa, T. Yanagita, "Randomness and Genuine Random Number Generator with Self-testing Functions", [Online], Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo, Hitotsubashi Memorial Hall Tokyo, October 2010, Available: http://www.letech-rng.jp/SNA+MC2010-Paper.pdf.

[14] L. Vokorokos, N. Ádám, A. Baláž, J. Perháč "High-Performance Intrusion Detection System for Security Threats Identification in Computer Networks", Elfa s.r.o. 2009, ISBN: 978-80-8086-131-5

Processing of multiple configuration sources using a dedicated abstraction tool

Milan Nosáľ
Department of Computers and Informatics,
Faculty of Electrical Engineering and Informatics,
Technical University of Košice,
Letná 9, 042 00 Košice, Email: milan.nosal@tuke.sk

Jaroslav Porubän
Department of Computers and Informatics,
Faculty of Electrical Engineering and Informatics,
Technical University of Košice,
Letná 9, 042 00 Košice, Email: jaroslav.poruban@tuke.sk

***Abstract*—Configuration is important part of design of software systems. There are many different configuration formats, such as XML, YAML, attribute-oriented programming, etc., that allow provider to design configuration language according to his requirements. Often target group of system users is very wide and one configuration language can not meet requirements of all users. The paper introduces and analyzes idea of supporting multiple configuration sources using dedicated tool. The tool provides common abstraction of configuration sources by creating virtual combined source. This paper presents design of such tool, based on analysis of existing tools. This design is extended with the idea of declarative representation of mapping of configuration languages to output format and of process of their combining. At last, paper presents proof-of-concept implementation of the tool called Bridge To Equalia and states several conclusions based on experimets realized with this implementation.**

I. Introduction

Using configuration in software systems allows configurable system to be modeled, customized or personalized to conform specific requirements of customer or to be adapted to special circumstances or environment.

Provider of the software system wants to utilize configuration to gain advantages in form of satisfying more clients, strengthening competitive advantages, providing more flexibility, robustness, quality and transparency in system. The system's ability to evolve is crucial to the user of system, and therefore it is to provider too(as his income depends on user's satisfaction). [1]

On the other sidestands user, that wants to customize system to meet her goals, preferences, abilities ands skills the best [2]. This way a configurable system can be universal enough but still satisfying users individuality and raising effectivity of her work [3]. Also, using a configurable system instead of custom one means spending less money, because customizing ready configurable system is usally cheaper than developing a new custom system (and also evolution of the system is easier to manage) [4].

A. Motivation

When implementing configurable system, its author has to design suitable configuration language for expressing concrete configurations of the system. Configuration as a concrete instance of configurable system is expressed as a sentence in a custom domain-specific language [5] (DSL), called configuration language. In designing process of language provider needs to consider many barriers and problems with adapting configurable system that are connected to used format of configuration language (for instance ones presented in [3], [6]). There are not merely technical aspects to consider. Each user prefers configuration language that meets her needs and taste the best.

During the design process of the suitable configuration language, there are considered aspects as simplicity of language, its verbosity and sententiousness, complexity of configuration process, domain abstraction, etc. [2], [6]. But user's preferences are based on subjective motives as well as on objective. It is easier to learn new XML configuration language than an annotations-based one, when you are familiar with XML but not with attribute-oriented programming (@OP is new and very popular format of configuration languages [7]). The wider the range of users, the more conflicts in requirements may occur.

To encourage user to utilize configuration, it is best to give her option of expressing configuration in the language, that is user most familiar with. The most straightforward way to deal with the problem is to choose one of available formats (such as XML, YAML, INI, @OP, etc.) and to design configuration language that best suits the needs of potential users. But usually there is not a format, that would meet all important requirements and its drawback would be negligible.

As an example can be considered Java annotations and XML documents. In Fig. 1 is listed a snippet of annotation-based configuration of Java Persistence API. The equivalent partial configuration in XML is listed in Fig. 2. We can see that annotations are less verbose. XML needs both opening and closing tags and duplicates names of source code elements (classes, methods, etc.). Annotations are easier to write as they are located directly on the source codes. This allows to configure simultaneously with programming. Configuration in annotations is tangled with sources, so it can not be Ïostšo easily (comparing to accidental deletion of configuration file). On the other hand, XML centralize configuration - whole configuration can be read much easier than by inspecting source codes. And another strong advantage of XML documents is, that changing configuration file does not need recompiling

program (changing annotations needs it).

```
@Entity(name = "Person")
@Table(name = "PERSON")
public class Person {

   @Id
   @Column(name = "ID")
   private String id;
   ..
```

Fig. 1. Snippet of annotation-based configuration of JPA

```
<entity class="pckg.Person" name="Person">
   <table name="PERSON"/>
   <attributes>
      <id name="id">
         <column name="ID" length="255"/>
      </id>
      ...
</entity>
```

Fig. 2. Equivalent snippet of XML-based configuration of JPA

There are also other aspects that complicate the decision for one configuration language. Configuration needed for localization of application can be as trivial as listed in Fig. 3 (using .properties files format). This configuration is not dependent on source code elements, and therefore it is not natural to define it through @OP. And imagine how cumbersome would be designing XML language for it (does not Fig. 3 feel more natural than Fig. 4?). Configuration structure is sometimes so simple, that using XML is unnecessary complicated (of course, when hierachy is needed, .properties are insufficient).

```
locale = en-gb
numberPrecision = 2
```

Fig. 3. .properties localization configuration

```
<local>
   <locale>
      en-gb
   </locale>
   <numberPrecision>
      2
   </numberPrecision>
</local>
```

Fig. 4. XML-based localization configuration

We came to conclusion, that differencies between available formats demand usage of multiple configuration language. Implementation of processing of more configuration languages costs more resources. Code, that process configuration, becomes larger, and its maintenance harder and error-prone (due to mixed processing of multiple configuration languages). There is also negative impact on evolution of used configuration languages, because changing language requires changes in processing code too.

Situation can be summarized in following two statements:

- **1** supported configuration language – higher risk of **dissatisfied users**.
- **Multiple** supported configuration languages – **increased costs** of implementation and maintenance of system.

This reasoning brings up a question: How to support multiple configuration languages without significant raise of costs and decrease of processing code simplicity?

II. Analysis

Let's suppose the system supports multiple configuration languages. Then it should to be able to process configuration in any of supported languages. In other words, if the system supports for example INI files and XML documents, user must be able to freely choose between these two formats. Second, stronger requirement is option of random mixing of configuration languages. The complete configuration is expressed as composition of partial configurations expressed in different languages. So far, we can identify three solutions: Ad-hoc solutions, source transformations and common abstraction of configuration sources.

A. Ad-hoc solutions

As the title suggests, provider needs to implement processing of all desired configuration languages ad-hoc. No dedicated tool is used. The concept of this solution is outlined in Fig. 5. Adding a new supported configuration language is basically implementing a whole operation of processing configuration in given language (difficulty is comparable to situation, when the system supports only one configuration language). But the fact, that the processing code needs to be integrated into existing configuration interface (code for other languages), makes the implementation even more difficult. The code processing one language may interleave with code for other languages, what results in worse maintanable implementation.

This approach is most common, but most inefficient too. As it is implemented in general-purpose language, it does not require working with new tool and developers feel freedom in defining a policy of mixing partial configuration into complete one. This approach can be recognized in many frameworks (e.g. Java Enterprise Edition, Microsoft Enterprise Library, GCore [8]), applications (Apache Tomcat), games (Fallout 2), and so on.

B. Source transformations

More elegant approach are source transformations. Instead of implementing processing for each configuration language, there is implemented processing of merely one language. For the other languages there are provided compilers (that can

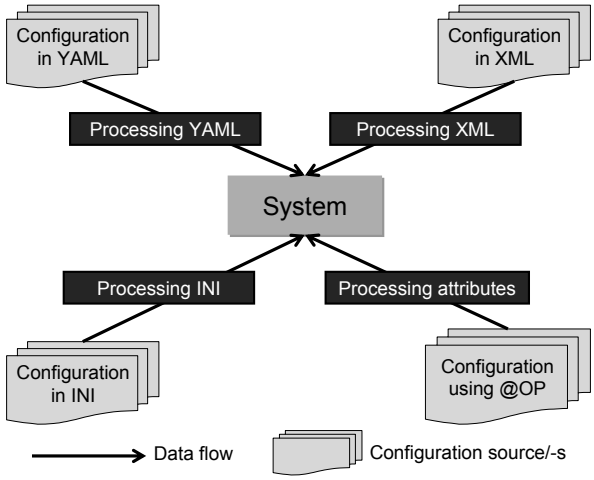


Fig. 5. Example of Ad-hoc solution with four configuration languages

be considered dedicated tools for translation) that translate configuration sources in unsupported languages to supported one. The concept is introduced in Fig. 6.

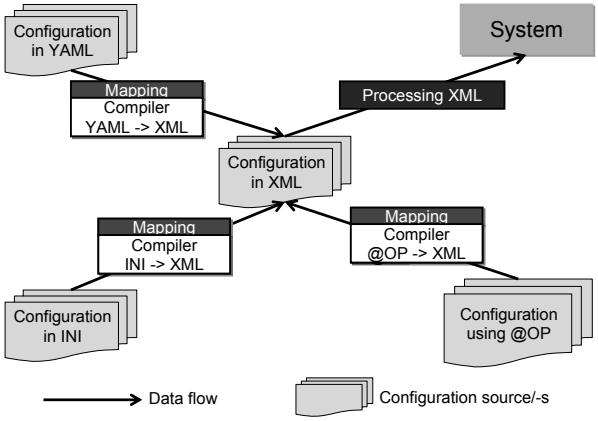


Fig. 6. Example of source transformations with four configuration languages

This way is manually implemented merely processing of one language, processing of other languages is transformed into translations between them and supported language. Description of translation is represented as mapping of one configuration language to another. Generally, translation description of language is shorter and therefore cheaper than implementing its processing. What's more, usage of compilers makes configuration processing code simpler (it processes only one language) and easier to maintain (as processing of each language is clearly separated from others).

The drawback of this approach is absence of support for combining (mixing) of configurations in multiple languages. Usually compilers just translate sentence from input language to output. And even with compilers capable of this functionality, the provider needs to deal with the synchronization of the translations to ensure proper priority of configuration languages. Fulfilling the stronger requirement for supporting multiple configuration languages is quite difficult.

C. Common abstraction of configuration sources

This paper suggests common abstraction of configuration sources as the solution to the problem. A dedicated abstraction tool is a program that processes the sources in multiple languages and by combining them generates complete model of configuration in output language as virtual configuration source in memory. This virtual source is used by configurable system for customization. Fig. 7 presents this solution. Complexity of mapping description between configuration languages and output language can be decreased if one of the input languages is chosen as the output language. This way resources needed to use this solution are equivalent as with source transformations.

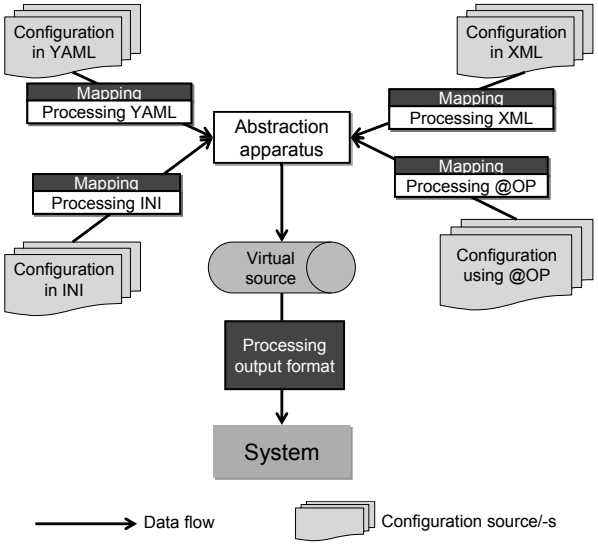


Fig. 7. Example of common abstraction of configuration sources

As examples of this approach can be mentioned Zend Config framework and Apache Commons Configuration, that abstract configuration in different formats (e.g. INI, XML, .properties). However, both of these lack effective means to define freely abstracted languages of supported formats. While Zend Config requires strictly default mapping between selected input formats (for instance, for one XML language there is only one supported language in INI files), Commons Configuration allows modification of default mapping in procedural way that appears to be too demanding for practical purposes.

III. Design of the tool

The drawback of supporting multiple configuration languages lies in larger and tanglier processing code. Using abstraction tool allows to deacrease code length and to clearly

separate processing of complete configuration (processing of virtual source) and processing of multiple languages (definition of language mappings). Problem with existing tools is that they lack sufficient support for different mappings between languages in multiple formats. The tool support only one default mapping between formats (as Zend Config does) or changing default mapping is too demanding (as in procedural definition in Apache Commons Configuration). This paper introduces a design of abstraction tool that tries to solve this issue.

A. Declarative approach to languages mapping definition

Of course, enabling change of default mapping between configuration languages is highly recommended, because it leads to greater freedom in configuration languages design and therefore raise the potential of satisfying individualities of tool's users. On the other hand, we find procedural representation of mapping (or changes in default mapping) and translation too demanding and therefore suggest utilizing declarative representation. As the concept of declarative representation can be considered a model of a model of a configuration, we call it metamodel. Metamodel describes how to create models of configuration in internal format from all input configuration languages and how to combine them into complete configuration. This way metamodel defines abstract syntax of configuration. It is important to dedicate enough time and resources to metamodel design, to make it as general as possible while remaining simplicity (in order to keep benefit of short representation of languages' mappings).

B. Conceptual design of the tool

Fig. 8 shows a conceptual design of tool based on analysis of existing tools and considering utiliziation of metamodel. Tool is composed of these modules:

- **Metaconfiguration reader**'s task is to process metaconfiguration - configuration of the tool. Metaconfiguration contains all necessary information for tool's operating. This module represents interface for system's provider to define required metamodel and other system-specific settings.
- **Metamodel generator** uses information mediated by metaconfiguration reader to build up in-memory representation of metamodel.
- **Configuration sources locator** takes care of locating and preparing sources for further processing. Purpose of designing this module is to separate preparation of sources from translation.
- When everything is prepared, configurations from multiple sources are translated into internal format using metamodel as a guide. This format unification is performed by **configuration formats unifying unit**. Result of this process is set of configuration models in internal format. Each of the model represents whole or partial configuration of the system.
- According to combining policy defined in metamodel (created by *metamodel generator* using metaconfiguration) the **combining unit** combines all models into one representing complete configuration. Result of this combining can be one of the models created by *configuration formats unifying unit*, if the model represents complete configuration and source format of the model has highest priority.
- In the end of the process comes to play **output module**. Its purpose is to return unified model in requested format (defined in metaconfiguration) to user of the tool. If requested format is not identical with internal, it has to perform additional translation.

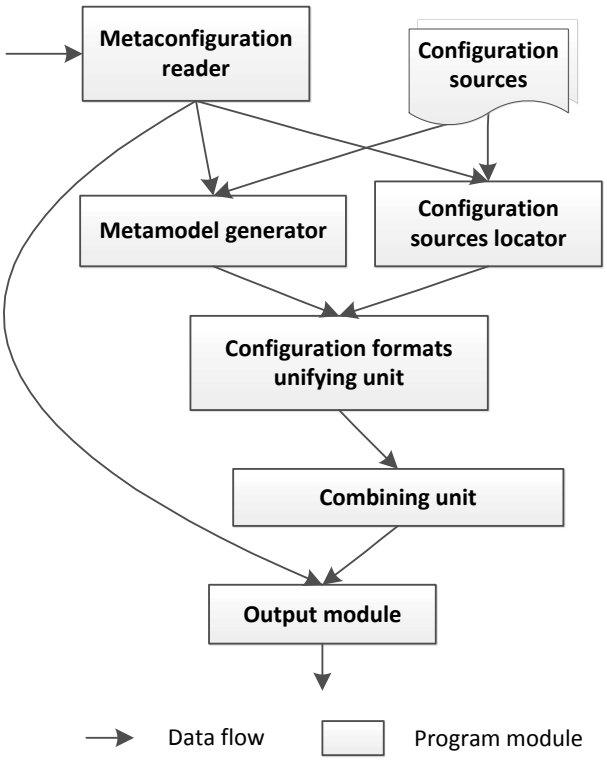


Fig. 8. Design of the tool

Modular approach brings better scalability of the tool, upgrading of the tool can be easily performed module by module.

IV. Experimental implementation

To provide practical basis for arguments stated in this paper, we implemented experimental tool and carried out few experiments.

A. Bridge To Equalia

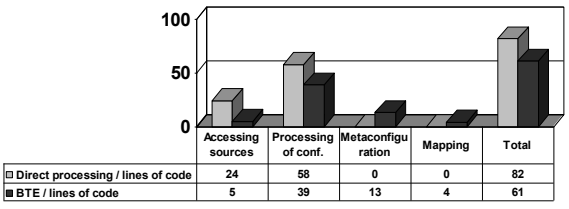
Bridge To Equalia (BTE) is a proof-of-concept implementation of abstraction tool. BTE is working with configuration languages based on Java annotations (implementation of @OP) and XML documents, two most commonly used configuration formats on Java platform. Of course, BTE introduces concept

of metamodel in order to make it easier to customize the tool for specific requirements of tool's users.

The tool is implemented in Java programming language. *Metaconfiguration reader* uses the tool itself to read metaconfiguration written through Java annotations and/or XML documents. For easier and more effective access to annotations, tool's *configuration sources locator* uses Scannotation tool. XML documents are read using standard DOM parsers. Structure of internal format is defined by custom Java classes. Object tree of their instances represent model of configuration. Metamodel is defined by Java classes too. It describes details of creation of configuration model in internal format from configuration in Java annotations or XML documents. Default metamodel is created according to annotation types that define configuration language in annotations. This metamodel can be altered to customize tool's behavior (modification is done by metaconfiguration).

We performed few experiments to test tool's flexibility and usability. Implementation of *metaconfiguration reader* module using the tool itself allows user to define metaconfiguration of BTE using both XML documents and Java annotations. This can be considered as proof of usability of the tool. To test flexibility, tool was used to process configuration of several Java EE technologies, that use both XML and Java annotations as configuration sources formats. Tests were performed for parts of Servlet, Java Server Faces and Java Persistence API configuration and all were successful (meaning that tool was capable to fully abstract both sources).

At last, we realized an experiment to compare direct processing of configuration to BTE-mediated. Its purpose was to show decrease of code length and to confirm assumptions about usage of abstraction tool. Not only the code was simpler (code processing configuration had to deal merely with XML instead of XML and annotations) but it was shorter too (Fig. 9). As the analysis showed, the benefit would be even more notable in less trivial case than used example.



	Accessing sources	Processing of conf.	Metaconfiguration	Mapping	Total
Direct processing / lines of code	24	58	0	0	82
BTE / lines of code	5	39	13	4	61

Fig. 9. Comparison of direct and BTE-mediated configuration processing

So far, BTE can be considered the most flexible and general tool abstracting XML documents and Java annotations.

B. Conclusions of experiments

Experiments led to following conclusions:

1) Significance of the benefits of using abstraction tool rise with increasing number of supported configuration languages. This is natural conclusion, abstraction of each language saves some code length and therefore more languages means more saved code.
2) Let's suppose that we have given number of input configuration languages. Larger abstract syntax of the configuration induce more complicated processing (one has to process more semantically different information) and therefore more code for each supported language. This results in more code saving.
3) Supposing we have given number of configuration languages, greater distance of language mappings from default induce metaconfiguration growth. This is due to bigger effort needed in describing changes in default mapping. Of course, larger metaconfiguration means less effective usage of the tool. Therefore it is very important for tool's author to find most common mapping between formats supported by tool and to use it as default.
4) Metamodel allows easier and faster changes to default tool's usage (in comparison with procedural expression used for instance in Apache Common Configuration).

To sum it up for practical purposes, using abstraction tool instead of ad-hoc solution is suggested in case of more supported configuration languages and configuration with larger abstract syntax, while languages are mapped to output format (or internal, depending on metaconfiguration policy) using default mapping or one close to default. On the other hand, with simple abstract syntax of configuration but complicated mapping of languages to output format, the tool's usage is not recommended (although it can make further extensions easier). It means, that using the tool should be carefully considered in design phase of software development, especially in development of small systems.

V. Current status and future perspectives

As was mentioned before, BTE is a proof-of-concept implementation of the tool. Its primary purpose was to test the concepts of source abstraction and declarative approach to language mapping. Although the tool was tested on aforementioned experiments, it would be necessary to put it through complex testing and debugging process before using it in "real-life" projects. Performed experiments where designed merely to confirm theses about importance of common abstraction of configuration sources and declarative approach in language mapping.

As next step in this area we see possible movement of metamodel definition from instance variables of metamodel classes to custom annotations. This way, metamodel would not consist of premade Java classes, but definition of metamodel would be totally upon the tool's user. Instead of defining configuration language (as it is currently in BTE, user defines annotation-based language and its mapping on XML), user would write directly abstract syntax of configuration languages in form of custom classes (tree of their object represents an abstract syntax tree). In other words, user would define, what is configurable in his system without concerning about concrete syntax of configuration languages. Concrete syntax

with combining policies would be added later in form of Java annotations.

VI. Conclusion

This paper concerns about enabling configuration from multiple sources. Its main purpose is to show and explain importance of common abstraction of configuration from multiple sources in comparison with other approaches. Paper presents new approach in designing abstraction tool with emphasis put on declarative way of defining mapping of abstracted configuration languages to output language. This design targets problems with tools usage caused by extensive configuration of the tool (in cases, when mapping between abstracted configuration languages is too far from default mapping supported by the tool). The effect of this approach is proven by experiments performed with proof-of-concept implementation of the tool for abstracting Java annotations and XML documents.

Acknowledgment

This work was supported by VEGA Grant No. 1/0305/11 Co-evolution of the artifacts written in domain-specific languages driven by language evolution.

References

[1] K. Bennett, P. Layzell, D. Budgen, P. Brereton, L. Macaulay, and M. Munro, "Service-based software: the future for flexible software," in *Proceedings of the Seventh Asia-Pacific Software Engineering Conference*, ser. APSEC '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 214–. [Online]. Available: http://dl.acm.org/citation.cfm?id=580763.785797

[2] B. Hui, , S. Liaskos, and J. Mylopoulos, "Requirements analysis for customizable software goals-skills-preferences framework," in *Proceedings of the 11th IEEE International Conference on Requirements Engineering*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 117–. [Online]. Available: http://dl.acm.org/citation.cfm?id=942807.943922

[3] W. E. Mackay, "Triggers and barriers to customizing software," in *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, ser. CHI '91. New York, NY, USA: ACM, 1991, pp. 153–160. [Online]. Available: http://doi.acm.org/10.1145/108844.108867

[4] H. C. Lucas, E. J. Walton, and M. J. Ginzberg, "Implementing packaged software," *Manage. Inf. Syst. Q.*, vol. 12, pp. 537–549, December 1988. [Online]. Available: http://dl.acm.org/citation.cfm?id=58996.58998

[5] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Comput. Surv.*, vol. 37, pp. 316–344, December 2005. [Online]. Available: http://doi.acm.org/10.1145/1118890.1118892

[6] P. H. Gross and M. J. Ginzberg, "Barriers to the adoption of application software packages," *Information Systems Working Papers Series*, vol. 4, pp. 211–226, january 1984.

[7] R. Rouvoy and P. Merle, *Leveraging Component-Oriented Programming with Attribute-Oriented Programming*. Karlsruhe University, 2006, vol. 2006–11, pp. 10–18. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.4416

[8] E. B. Passos, J. W. S. Sousa, E. W. G. Clua, A. Montenegro, and L. Murta, "Smart composition of game objects using dependency injection," *Comput. Entertain.*, vol. 7, pp. 53:1–53:15, January 2010. [Online]. Available: http://doi.acm.org/10.1145/1658866.1658872

Source file copyright protection

Eva Danková
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics,
Technical University, Košice, Slovakia
eva.dankova@tuke.sk

Norbert Ádám
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics,
Technical University, Košice, Slovakia
Norbert.adam@tuke.sk

Peter Fanfara
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics,
Technical University, Košice, Slovakia
peter.fanfara@tuke.sk

Marek Dufala
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics,
Technical University, Košice, Slovakia
marek.dufala@tuke.sk

***Abstract*—Study is dealing with the suggestion of copyrights from the different points of view. It is concerned with copyrights which protect source text files from their misusage (copying). It explains legal differences in protection of source text files in Slovakia and other countries EU. It marks basis of cryptography and its standards. These standards are used in the suggestion of preventive system and also in the suggestion of authenticity system. The outcome is independent application under operating system Windows which can partially code source text files.**

Keywords-component; copyright, protection, code rewrite,

I. INTRODUCTION

Computing has brought fast pace of development in technical research sectors and gradually penetrated into the everyday life of people. Every research and development requires a degree of confidentiality of the acquired knowledge, and there for the demand for secrecy contributed to the rapid development of cryptography. It was established standards for cryptographic algorithms, which do not offer complete coverage of the needs for confidentiality.

One possibility to protect source code is copyright protection. Copyright is a form of protection provided by the laws to the authors of "original works of authorship," including also software programs. A copyright gives the author the exclusive right to a work for the life of the author.

Today's time is the time of technology and computer, and in everyday life, as work, hobby, communication, is used some kind of software. Therefor is software a very good business today, so authorship protection of source files is a fairly important area.

In this paper are described some protection possibilities and finally the design of the system. This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10. And also this research was supported by VEGA 1/0026/10.

II. CRYPTOGRAPHY - DATA CONFIDENTIALITY

Message confidentiality results from the knowledge that the message can be capture during the transfer. For this reason, from the beginning of the transfer, was developed message transmission methods for concealing the communication, which can be divided into two groups:

- Steganography
- Cryptography.

A. Steganography

Steganography use concealing methods for communication based on hiding messages so, that the secret message transmission takes place behind the non-confidential communications. It is sheltering technology which keep message inside a message, respectively hide secret information in the message. According to used methods, can steganography be divided into:

- Technical steganography
- Linguistic steganography.

Technical steganography use for classification or hiding secret messages technical procedures (invisible inks, concealment of transmission media, respectively, extreme reduction of its dimensions, etc.). Technical steganography survives in various forms (such as microdots) until today, where its evolution follows the modern technical procedures and provides a specific degree of data secret.

Linguistic steganography use text or other graphical form to hide a secret message. Better form of linguistic steganography is hiding message in a random positions of covering text. Hidden message can be obtained by using the agreed grid with gaps on these positions. The aim of steganography is to realize a secrete communication in a background of public communication that takes place in electronic form.

Steganography in the modern sense is a part of a larger area, which is known as information security. Transmission of

hidden information takes place in subliminal channels that can be created in the existing communication networks using modern steganography methods.

B. Cryptography

Cryptography does not aim to keep secret the existence of messages, but to keep secret its information content using encryption method [3]. Encryption is a process of adapting message with the aim keeping the content secret. After message encryption, unauthorized person shouldn't be able to get the information content from the original message. The specific encryption procedure is also known as encryption algorithm or cipher. Encryption requires two components:

- Encryption algorithm
- Key.

The encryption algorithm is a procedure (rule), which generates an encrypted message. The key is additional information that is necessary for the implementation of encryption algorithm. The basic thesis of cryptography by Kerckhoffs is that the security of encryption can't be based on secret encryption algorithm, but only on the secret key. Classical cryptography, which is known as secret key cryptography use two basic approaches, which are:

- Substitution - replaces each symbol of the original message by another symbol, which remains in the same location as the original symbol. The choice of symbol substitution determines the key.
- Transposition - the symbol arrangement of the original message is in another way (e.g., permutation), which is determined by secret key.

The development of computers and electronics has brought new forms of encryption algorithm implementation, but it has also improved the effectiveness of cryptanalysis, respectively systematic review of encryption algorithms security. According to development of computers was established a conception cipher security, which is given by current state of computer technology and by development state of efficient computational algorithms [3].

III. SYMMETRIC ENCRYPTION

Cryptographic systems with a secret key use the same key for encryption and decryption, and are also called symmetric, respectively conventional cryptographic systems. The safety of these systems is in the secret key that the sender and addressee must exchange before the communication. This exchange is a disadvantage in terms of emergency communications. Symmetric ciphers can be divided into:

- Block symmetric cipher – realize encryption or decryption in a blocks,
- Stream symmetric cipher – realize encryption or decryption by symbols.

A. Stream cipher

Stream cipher transforms each symbol from open text p, to the symbol of the corresponding encrypted text c. The simplest implementation of a binary stream cipher is shown in the following figure:

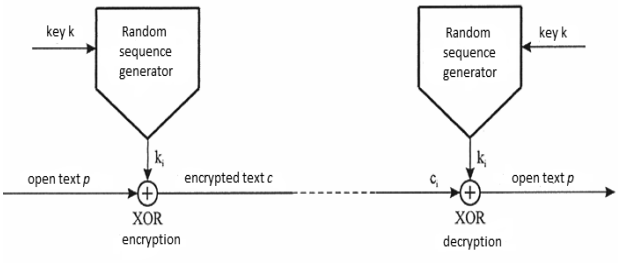


Figure 1. Principe of stream cipher

Pseudo-random sequence generator generates a pseudo-random binary sequence $k_1, k_2, k_3,..., k_i$, which is dependent on the key k. Encryption is implemented through XOR binary operation symbols p_i of the open text, to give the binary symbols of encrypted text. So it gilt:

$$c_i = p_i \text{ XOR } k_i \text{ pre } i= 1,2,3,... \qquad (1)$$

Decryption is implemented in analogy to the relationship:

$$p_i = c_i \text{ XOR } i_i \text{ pre } i= 1,2,3,... \qquad (2)$$

where during the decryption must be used the same pseudo-random binary sequence $k_1, k_2, k_3,..., k_i$, also the same pseudorandom generator with identical key k. A little resistance to integrity disruption of the encrypted message follows from the principle of stream cipher, where the encryption takes place by open symbol text. If the cipher is broken, an unauthorized person can modify the transferred message, without the receiver knows it.

B. Block cipher

Block ciphers transform a group of symbols from the open text to a symbol group of encrypted text. So the encryption is realized by blocks of the clear text. Block size is usually not important. Symbols from open text are grouped into a block, on which is applied a cryptographic transformation E_k dependent on key k. The result of encryption is encrypted text blocks that have the same size as an open text block. Decryption is based on analog procedure. On encrypted text block is applied the inverse cryptographic transformation D_k, which uses the same key as the E_k. The result of this process is obtaining the original clear text block. Safety of block ciphers can be enhanced in several ways, which adapt described basic process of block ciphers in order to complicate their cryptanalysis. The basic procedures include:

- Multiple encryption
- Doubling the length of the block
- Encryption in cycles.

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10. And also this research was supported by VEGA 1/0026/10.

IV. NONSYMETRIC ENCRYPTION

Cryptographic systems with public key use one key to encrypt and second key to decrypt, so it gilt:

$$C = E_{k1}(M) \quad (3)$$

$$M = D_k(C) = D_{k2}(E_{k1}(M)) \quad (4)$$

Encryption key k_1 is a public key and decryption key k_2 is a private key known only to the addressee. Cryptosystems with public key is also known as asymmetric cryptosystems. Their security is based on the mathematical complexity of determining the private key and from known public key.

Public key cryptography is based on asymmetrical cypher, which use two keys. One key is used to encrypt, the second to decrypt, what is a fundamental difference in compare to symmetric encryption. In symmetric encryption is necessary to ensure the secrete distribution of keys to two communication parties. The encryption data exchange process can be initiated only when the secret key distribution is completed. Distribution of key in symmetric encryption deliver problems mainly due to secure key distribution channels classified, which imagine weaknesses in the concept of symmetric encryption.

Big advantage cryptography with public key is that it doesn't require interaction between the parties before start the exchange of encrypted text or data. Each participant in a cryptographic system with the public key has two keys. One key that is known as private key, the second is known as public key. The private key is secret and public key can be published. Using two keys affect the security level, method of key distribution and also user authentication. Public key cryptography is universal method and allows to realize basic functions such as message content confidentiality, user and data authentication.

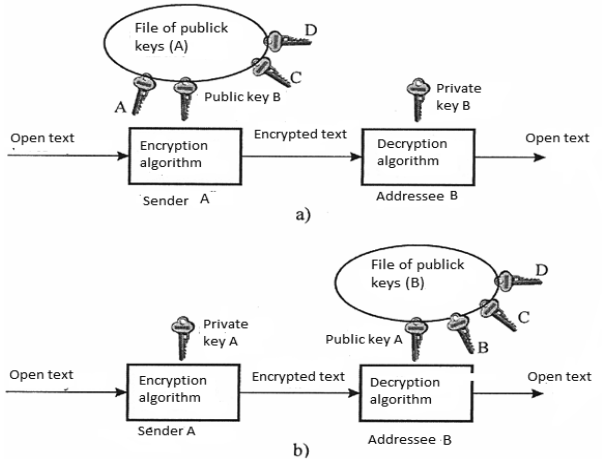


Figure 2. Publick key cryptography , a) encryption, b) decryption

Principles and functions of cryptography with public key are given in Fig. 2. Public key cryptography can provide following functions:

- Encryption
- Authentication.

Both features are based on the fact, that each participant owns its communications private key and public key. Both keys built a pair, where the private key of each participant is secret and public key is available to each participant.

V. AUTHENTICATION

Cryptography with public key enables to solute problem with secure communication and secure key distribution. But it also brought the problem of user and data authentication, which is one of the most important problems of network security. User authentication can be characterized as a process of identification and verification. User authentication in cryptography is essential especially with regard to potential attacks such as faking identity, respectively forged public key.

User authentication and data authorization is sometimes referred as message authentication, which can take place at two levels. The lower level of authentication includes application authentication mechanism, which based on the message produces a certain amount, known as authenticators. A higher level includes the generation of authentication protocol, which uses authenticators and which perform the actual authentication messages.

VI. PROPOSAL OF SYSTEM PREVENCION

Prevention system is a software solution for confidentiality of information against unauthorized access without damaging the functionality of the source files. The system consists of several sub-operations, which are divided into different groups according to different perspectives. As mentioned in the case of encryption, it must be count with a specific disadvantage. Individual systems, which use special text formats, allows some form of uncertainty and some sections specifically marked, that are not processed. Source files of several languages, which are intended to a form of logic, after text transformation, provides sufficient space to programmers (functions, classes, etc.).

Prevention system uses this latitude and enabled structure of source files to transform the input files. During the transformation are used such measures (encryption) that preserve the functionality of the source text file. System output can be text files, which are not of the program nature, a special encrypted file and encrypted source text files. Ideal encrypting for the source text file is encryption with no loss of functionality, this means that the encryption source code does not lose its functionality, so you can recompile the encrypted file to the same output (the executable program, module, library, etc.), as the original. It is true, that the encryption algorithm can't change concrete parts of the source code. This type of encryption is based on changing parts of open source file, of which change doesn't damage functionality of text source file. The proposed software solution is from the graphical site divided into three main sections (Figure 3.).

- Basic,
- Medium,
- Professional.

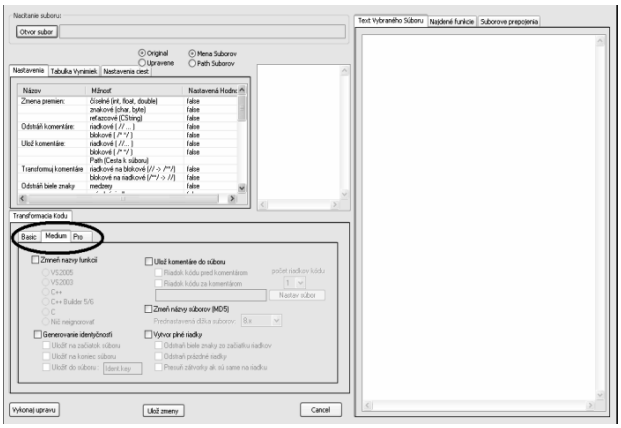


Figure 3. User interface of prevention system

A. Basic part

Changes, which belong to basic in terms of copyright prevention, are characterized by its typical feature. Namely it is that after changing the basic structure of the open text doesn't change, with the exception of possible removing of comments sections. The aim is to achieve the minimum level of readability and understandability without harming performance. This aim in the meaning of basic changes can be achieved in three ways.

- Editing comments
- Transforming of variable names
- Compressing the text.

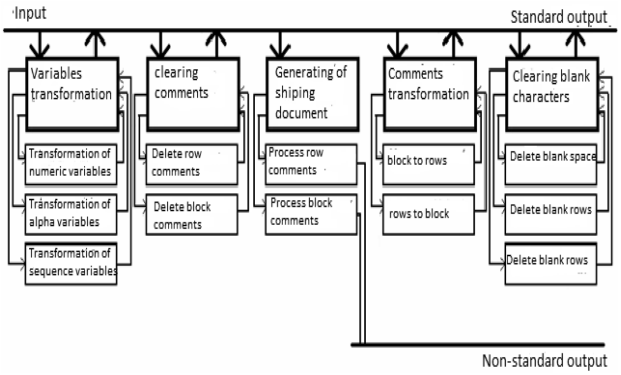


Figure 4. Block diagram of the interface of basic prevention system

Editing of comments - under term editing of comments you can imagine countless variations of simple changes (from changing small letters on large to their deleting). Editing comments can be divided in:

- Transformation
- Export
- Removing.

Transformation of variable names - One of the biggest parts of discretion writing source files is the declaration of variables. The basic style of encryption software is based on this basic pillar of readability (suitably chosen variable names). Suitably chosen names of variables give unimaginable lead to the understanding of the source code. Based on previous conditions, it is best to encrypt the variables names using the hash algorithm with the steady-state control.

Compression of the text - any text can be treated by removing the white (invisible) characters, if it does not damage the functionality. Not every white character can be removed without damage. Continuous text without spaces would give the impression of an encrypted form, but it would have violated the basic idea of encryption without damaging functionality.

Using the proposed solution is specified for programmers and provides some protection against plagiarism, and some intellectual property protection. To demonstrate the proposed solution was developed a software solution in the form of "prevention system". Basic steps of preventing are designed for computer confidentiality against unauthorized access. Individual steps can be executed separately and offer some form of solution.

B. Medium part

Changes belonging to medium in terms of preventing copyright protection are characterized by enlargement of the basic interface, with offer to the above authentication systems and with intervention to the names of source files. The aim is to achieve a higher level of readability and understandability than basic interface without damaging the functionality. This aim e in terms of extended treatment can be achieved in the following ways.

- Transformation of functions and classes names
- Support for authentication
- Compressing text
- Handling of file names
- Enhanced support for the accompanying document.

Transformation of functions and classes names - Transformation of this nature is very courageous to express in general, therefore is the proposal focused on rules for the C language [4].

Transformation of function names and classes has certain rules, which must be set in advance, if we want to preserve the functionality of the source text file.

- 1. Rule - Be able to differ, which classes and functions belongs to libraries, those don't flow over implemented changes.
- 2. Rule – It is necessary to know the absorption of files, where the changes are executed. And that for the reason to modify also the calls of changed classes and functions.

- 3. Rule – Don't to change directives and commands of the development environment, which must remain in the same form.
- 4. Rule - Changes must be made global according to coherence of the source files.

By keeping these rules is a small chance that by changing of classes and functions names will be damaged the functionality of the source text file. With use of this change will be achieved high privacy of complicated solutions contained in multiple files.

The possibilities of advanced interface offer extensive opportunities for confidentiality source text file. The assumption is bound to use of the basic interface. The extension is necessary for the authentication system, because of its offer of generating authentication code.

C. *Proffesional part*

Changes belonging to a professional in terms of copyright prevention are characterized by special treatment in the source text file, in the form of specific acts of shape changes in the structure of source text file, by adding special parts with the exception of auto extractor.

The aim is to keep the parts of the text secrete by duplicating fragmented parts and to achieve a higher level of readability and understandability than previous interfaces without harming functionality.

This objective in terms of extended treatments can be achieved in the following four ways:

- Increasing of open text by additional comments
- Adding blank pieces of code
- System „auto extractor"
- Changing key pieces of code

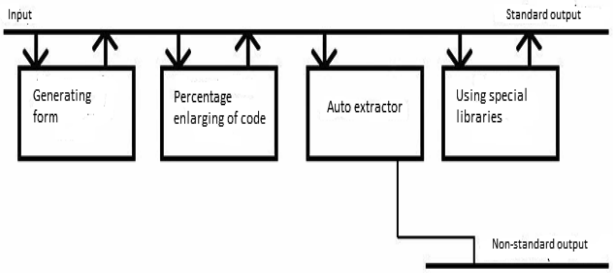


Figure 5. Block diagram of the interface of professional prevention system

Options of professional interface offer for source text files sufficient confidentiality. The assumption is bound to use of the basic interface and enhanced interface. Professional interface is oriented in extending the code to add different parts of the source text file and to offer a special privacy for the transfer of the various transmission systems.

VII. USER AUTHENTICATION

Owner authentication is very important, especially with regard to potential attacks, such as faking identity [3]. Copyright protection in the proposed system is governed by a similar system, such as patent protocol. The owner of the authorship code becomes the person who first introduced source code into the system.

Solution is provided as authentication system that would contain the source code stored in databases in a form of encrypted tables, from which you can't create the original files.

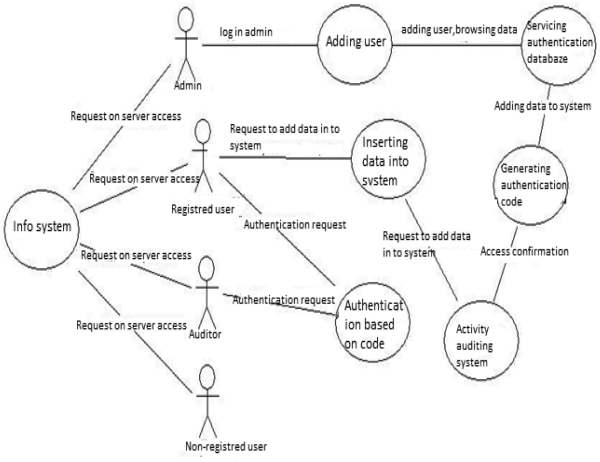


Figure 6. Authentication principle

System is divided into 2subsystems according to the main activity:

- Authentication subsystem.
- Generating system of authentication data.

Authentication system is a system based on obtained data from the user based on which the generating authentication system of data generates a series of codes. Authentication system uses the generated data and compares them with data from the central database based on the input authentication code, which is a part of the input data from the user.

Authentication data generation system is aimed at transforming the input data to output authentication table. The system recognizes two types of input data:

- Source text files,
- Non-text format files.

Authentication system has today reasonably use. It offers protection against plagiarism, against the possible misuse of the source files and also it encourages creating better and improved products. On user site it offers a specific form of protection, because it undertakes author to product liability.

The system can also be used in education to check the similarity of the works written in the form of text source files. This system is designed so that the similarity of source files is revealed in the form of percentages.

VIII. CONCLUSION

Copyright protections of source files and their violation is currently not solved and over time an increasing problem in the information society. In the age of electronic communication (Internet), there is frequent devaluation of the individual. The proposed solution of "prevention system" is an effective tool to deal with the problem of plagiarism. The proposed tool is based on the transformation of the source files on multiple levels. The different levels are interlinked and ensure the confidentiality of the source text file to a different extent.

The actual use of expanded or professional level to transform the source text file has a lower classification as a separate force using only a basic level. But gradually, using all levels of prevention is multiplying the percentage of classification (growing exponentially). An important part of prevention is to support authentication in the form of generating authentication code. In fact, proposed system is based on authentication system, which is based on the principle of patenting new manufacturing technologies and offers a possible solution for a software patent.

The proposed solution is largely linked with the rules of creating source files in C and C + +. Other development standards may be some clashes with that procedure, thus extending the systems can support different programming languages.

The use of similar systems is quite extensive in education and in different sectors, which are involved in the programmer's activity. The introduction of the proposed system would significantly reduce the development of plagiarism.

ACKNOWLEDGMENT

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10. And also this research was supported by VEGA 1/0026/10.

REFERENCES

[1] Meško, D., Katuščák, M. a kol,: Akademická príručka. Publishing house: Osveta, Martin, 2004. ISBN: 80-8063-150-6.

[2] Singh, S.: Kniha kódu a šifier. Publishing house: Dokorán a Argo, Praha 2003. ISBN: 80-86569-18-7.

[3] Levický, D.: Kryptografia v informačnej bezpečnosti. Publishing house: elfa, s.r.o., Košice 2005. ISBN: 80-8086-022-X.

[4] Herout, P.: Učebnice Jazyka C 3. Neat edition. Publishing house : Kopp, České Budějovice 2001. ISBN 80-85828-21-9.

[5] Eckel, B.: Myslíme v jazyku C++ knihovna programátora. Grada Publishing, s.r.o. Praha 2000. ISBN 80-247-9009-2.

[6] Kruglinski D. J.,Sepherd G., Wingo S.: Programujeme v microsift visual C++.Computer Press Praha 2000, ISBN 80-7226-362-5.

[7] Oberholzer-Gee F., Strumpf K..: File-Sharing and Copyright, .Harvard Business school 2009, Work paper.

[8] Vokorokos L., Kleinová A., Baláž A.: Architecture of data security with the aspect on the intelligent control systems, In: Energija-ekonomija-ekologija. Vol. 10, no. 3 (2008), p. 031-035. - ISSN 0354-8651

[9] Feik M..: Vykrádanie nápadov alebo brzdenie inovácií?, http://www.blisty.cz/art/19081.html

[10] Meško D..:Plagiátorstvo, In: http://www.etd.sk/doc/plagiatorstvo.doc.

Traffic sign recognition based on the Rapid Transform

Jan Gamec, Daniel Urdzík, Mária Gamcová
Dept. of Electronics and Multimedia Communications, Technical University of Košice, Park Komenského 13,041 20 Košice, Slovak Republic
jan.gamec@tuke.sk, daniel.urdzik@tuke.sk, maria.gamcova@tuke.sk

***Abstract*—In this paper is presented a model of system for invariant object recognition, which consists of five stages. The first stage shifts the object so that the centroid of the object coincides with the center of the image plane. The second stage is an application of the polar-coordinate transforms used to obtain N-dimensional vectors-representations of the input object. In this stage, any rotation of the input object becomes a cyclic shift of the output value of this stage. The third stage employs CT (Certain Transform) a class of shift-invariant transformations to provide invariant representations for cyclically shifted inputs. The next stage normalizes the outputs of the previous stage to obtain scale invariance. In the final stage is realized a classification.**

Keywords-symbol recognition, rapid transform.

I. INTRODUCTION

Ability of symbol detection and recognition in an image, independent from its position, size and orientation – otherwise invariant symbol recognition ability is very important aspect of automated symbol recognition. Many techniques developed for this purpose are based on existence of available library, which contains saved images (features) and recognition problem rests in searching for match between one of these features and symbols in the image. Searching for matches means comparison of features characteristics file and characteristics of input image. This procedure can take very long time, due to huge amount of input information, as well as information about known symbols (features). Thereby, very important task is to build up a system of detection and symbol recognition, which reduces this amount and which is able to reach fast recognition.

One of options how to carry it out is application of transformations of an input image and consecutive characteristics (features) selection for next processing which appropriately represent the symbol. Process of recognition based on extraction of features can be divided into several simpler processes:

- Acquiring of input data intended for recognition in form, which allows next processing (digitalization, preprocessing).
- Input information analysis: symbol detection in input image, extraction of features, which characterize input symbol.
- Classification: input symbol recognition, i.e. class definition of known (learned) symbols, which characteristics is mostly matching with characteristics of unknown symbol using selected criteria.

Neural networks were successfully used at pattern extraction which is used for object recognition. Within the technology of neural networks, there is a group of images containing known objects placed into the pattern library. Network extracts characteristics from this group of pattern pictures, which are later used for comparison with characteristics of extracted unknown objects images. Despite of the results reached in this area and existence of many various structures of these networks, their practical use has limitations in many cases. Most of already existing networks require many connections and processing units, which amount are rapidly increasing within enlargement of input image, due to invariant object recognition. For example invariance towards rotations and object size change can be reached in many cases only by integrating rotated and enlarged versions of these objects into the group of pattern images [1], [2].

From given reasons we can say that symbol recognition invariant on shift, rotation or scale variation isn't trivial problem. Within the process of recognition, there are used fast translation invariant transform of the class CT (Certain Transform). Submitted model is intended for binary images and thus system is able to process large image areas.

II. MODEL OF THE SYSTEM FOR INVARIANT SYMBOL RECOGNITION

Model of the system for invariant symbol recognition consists of five levels each of them has set its own specific function. Model of the system for invariant symbol recognition consists of five levels each of them has set its own specific function (Fig.1). In the first level symbol in the input image area shifts to the centroid of the symbol and matches the display area. In the second level coordinates are converted into polar ones by polar transformations. Outputs of this level are generally N-dimensional vectors which characterize the symbols. Operations accomplished in this level cause that any output symbol rotation is expressed by the cyclic shift of outputs of this level. In the third level scale factor is removed from the outputs of the previous level by normalizese of these outputs to obtain the invariance to change the size of the

symbol. In the next level fast translational invariant transformation (rapid transform, modified rapid transform or other CT class transformation) which provides invariant representations of cyclically shifted inputs is applied on the results of the third level. The last level is the classification network whose purpose is to identify the unknown input symbol based on selected criteria.

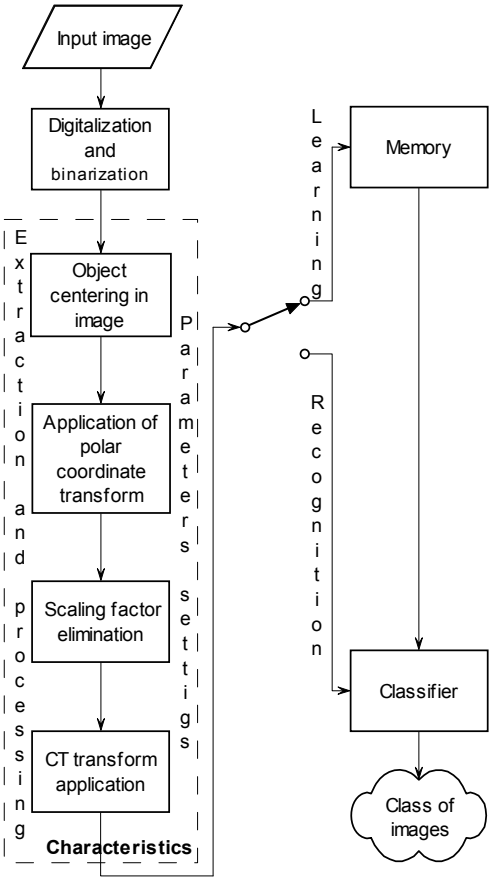


Figure 1. Block diagram of the system for invariant symbol recognition

A. Symbol centering in the input image

Elimination of the information about position of an symbol in the input image, or to do operations with input image, which cause the information to be not essential, is needed for succeeding in symbol recognition invariant on shift of the symbol [1], [5]. This process is running in the first level of described model of the system by centering of the symbol in the input image that means symbol is shifted, so its centroid is matched the center of image area. Module of discrete Fourier transform shows the invariance to shift too, despite that DFT usage in this level isn't appropriate for next reasons:

DFT needs complex arithmetic operations, by what is network complicating.

Phase spectrum DFT is not invariant on shift and so it's not possible to use it in this level what cause loss of the information about symbol, which DFT phase spectrum doesn't contain.

Symbol centering network is more preferable in comparison with the DFT to use for acquiring invariance to shift, which has quite simple implementation and which saves all information about input symbol.

Subsystem for symbol centering contains two directional sub-networks: shift network for horizontal direction and shift network for vertical direction. Each of these sub-networks contains K lower levels and on each of these levels, there is a symbol shifted by distance (in image elements – OP) equal to second power. This also means that on any j-*th* lower level is not the symbol shifted by distance 2^j OP or any other shifting. Function of this level, shifting the symbol in vertical direction by 2 OP.

Symbol management calculate coordinates of symbol centroid in actual input image, which are needed for calculation of relating shifts in directional sub-networks of subsystem for symbol centering. Coordinates of the symbol centroid $(\bar{x}, \bar{y})$ can be calculated by formulae:

$$A = \sum_x \sum_y f(x, y), \qquad (1)$$

$$\bar{x} = \sum\sum x.f(x, y)/A, \qquad (2)$$

$$\bar{y} = \sum\sum y.f(x, y)/A, \qquad (3)$$

where $f(x, y)$ is value of OP with coordinates (x, y), which can be 0 or 1 (binary images) and A is number of image elements with value 1 or can also express area (scale) of the symbol. After finding the $(\bar{x}, \bar{y})$ required shifts are calculated, which need to be done, for matching the symbol centroid to the center of image area. These shifts are later encoded into binary codes, based on which are for each lower level of directional shift networks generated by management circuit three managing signals. The signals are *U*, *N* a *D* (Up , Down and Neutral), which can get value "0" or "-1". Value "0" means, that managing signal is active and value "-1", that signal is inactive. For example, if it's needed to shift the symbol upwards in mentioned lower level, then managing signals have values: $U = 0$, $D = -1$ a $N = -1$.

Function of directional networks is similar to binary codes transfer mechanism into decimal representation. After calculation of values required for shifting in mentioned direction, will be whole operation of shift realized on K lower levels of the shift network for this direction. Centroid of the symbol doesn't have to be integral, but then its coordinates aren't going to be matching the center of the image area. However, if the symbol is big enough, then this inaccuracy won't have very large impact on result of the recognition.

B. Application of a polar coordinate transforms

On purpose of analyzing the symbol rotation around the axis going through its centroid, are in the second level of the described model applied polar coordinate transforms on

Identify applicable sponsor/s here. If no sponsors, delete this text box. (sponsors)

outputs of the first level [1], [2], [3], [4]. Transformation of the information about the symbol into axis θ and r allows extracting of such a characteristic, within which rotation of the symbol affects as cyclic shift of their values.

General implementation of discrete polar transform has two dimensional structure of MxN points, organized as two dimensional space with axis r and θ. Value of the point $p(r, \theta)$ in polar plane is derived from corresponding value of image element (x, y). Each circle in image plane, independent from its scale, is represented by same amount of points N in polar plane. Because of the certain amount of image elements, which creates perimeter of a smaller circle is shown as higher amount of element in perimeter of big circle. For example circle with radius r=3 has perimeter 18 OP, so $p(3, \theta) = f(3, 0)$ for $-N/(2.18) < \theta < N/(2.18)$, where $f(3, 0)$ is value of image element (3, 0). Each point $p(r, \theta)$ can be acquired same.

Amount of N discrete values of angle θ needs to be chosen so, that within this transform were the least losses of the information about the symbol. For representation of the symbol are used characteristics – vectors, which reduce information included in MxN points of polar plane into N dimensional vectors set, enough characterizing symbol, where symbol rotation affects as cyclic shift of their values. Each elements of given vectors are calculated by formulae:

$$p_1(\theta) = \sum_r p(r, \theta), \tag{4}$$

$$p_2(\theta) = \sum_r r.p(r, \theta), \tag{5}$$

or

$$p_3(\theta) = \sum_r r^2.p(r, \theta), \tag{6}$$

etc.

Due to the similarity of these formulas with this one for calculation of moments are these vectors called moment of the rows.

Number of discrete angle θ values has great significance on this model's level. This parameter influences the quality of transforming process in terms of losing the information of the symbol and on the other side determines the size of entering characteristics i.e. degree of reduction of these information and so it determines the speed of the distinguishing process. Due to the facts, setting of the optimal value of this parameter is difficult and possible by setting up the experiment. Reaching the perfect recognizing is in most cases possible only by using characteristics – moments of zero and first orders. Sometimes is necessary to use moments of higher orders for increasing the recognition ability. Better recognition ability of these moments is resulting from their structure, where the values of each image elements are "balanced" by higher orders of relevant values of r (6), which causes their increased sensibility for distortion of the symbol. Influence of usage of these moments on the results of the recognition was subject of many experiments.

C. Elimination of scaling factor

Elimination process of influence of scale variance is based on next formulas. Integrals are used to simplify, in sum operations in formulas (4) and (5). Let:

$$\int_r p(r, \theta) dr = p_1(\theta) \tag{7}$$

and

$$\int_r r.p(r, \theta) dr = p_2(\theta), \tag{8}$$

then

$$\int_r p(r/n, \theta) dr = n.p_1(\theta) \tag{9}$$

and

$$\int_r r/n.p(r/n, \theta) dr = n^2.p_2(\theta), \tag{10}$$

where n is mentioned scale factor. By next application of logarithm on results of integrals:

$$\log(n.p_1(\theta)) = \log(n) + \log(p_1(\theta)) \tag{11}$$

and

$$\log(n^2.p_2(\theta)) = 2.\log(n) + \log(p_2(\theta)). \tag{12}$$

Resulting from these formulas we know, that by eliminating of constants $\log(n)$ and $2.\log(n)$ we acquire characteristics invariant on symbol scale shift. Removing of these constants can be reached by using:

$$p_i'(\theta) = \log(p_i(\theta)) - \max_\theta(\log(p_i(\theta))) + C, \tag{13}$$

where C is such a constant, where elements p_i' are always positive. In such a case, where $p_i(\theta)$=0, and $p_i'(\theta)$ is equal zero. It's not necessary to calculate the maximum value of $\log(p_i(\theta))$ in formula (13), because maximal element of vector p_i provides the previous level of this model. Realization of this model isn't very complicated.

D. Application of rapid translational invariant transform

Class of rapid translational invariant transforms (CT - Certain Transforms) [6] was derived from rapid transform (RT) by generalizing of operators pair $f_s(a, b)$ where (s=1, 2) in original algorithm . RT is rapid, non-orthogonal, non-linear, translational and partly rotational invariant transform. Transforms of CT can be divided by using operators $f_s(a, b)$. CT transforms are used within the processing of image information in robotics, in problematic of symbol recognition, can be used within estimating of movement [6] etc. One of their most important properties is speed, translational invariance, partial invariance on rotation and scale variance and simple technical realization. They were described in [6], [7], [8]. The most frequently used transforms of CT are in tab. I.

TABLE I. FREQUENTLY USED TRANSFORMS OF CT CLASS

	RT	NT	MT	QT
$f_1(a,b)$	$a+b$	$max\{a,b\}$	$a \vee b$	$(a+b)^2$
$f_2(a,b)$	$\|a-b\|$	$min\{a,b\}$	$a \wedge b$	$(a-b)^2$
Input values	*real*	*real*	*binary*	*real*

E. Modified rapid transform

Resulting from properties of CT class transforms, they are no able to recognize reversed symmetrical inputs and in the process of recognition nor the reversed symmetrical symbols. For curing this unwanted invariance, there was designed modified rapid transform (MRT -Modified Rapid Transform), which preserves property of invariance to translation.

Modification of original rapid transform was done by putting the k (in general) modified steps before calculation of RT itself.

Operator f_0, used in preprocessing MRT can be realized by next simple formula:

$$x'_i = f_0(x_i, x_{i+1}, x_{i+2}) = x_i + |x_{i+1} - x_{i+2}|. \quad (14)$$

It's important for removing of unwanted invariance to reflection of input data, that the operator f_0 is asymmetrical.

Two 1-dimensional MRT for each direction can be used for 2-dimensional input signal, or one 2-dimensional MRT. Goal of this modification is removing of unwanted invariance to reflection of input data. For 2-dimensional MRT can be used in steps of preprocessing next modified operations:

$$\begin{aligned} f_0^x &: x'(i,j) = x(i,j) + |x(i+1,j) - x(i+2,j)|, \\ f_0^y &: x'(i,j) = x(i,j) + |x(i,j+1) - x(i,j+2)|, \\ f_0^{x+y} &: x'(i,j) = x(i,j) + \\ &\quad + |x(i+1,j) - x(i+2,j)| + |x(i,j+1) - x(i,j+2)|, \\ f_0^{xy} &: x'(i,j) = x(i,j) + |x(i+1,j) - x(i,j+1)|. \end{aligned} \quad (15)$$

From consideration and computer simulations are resulting next results:

- MRT using f_0^x cannot suppress the invariance to reflection of input data in y direction.
- MRT using f_0^y cannot suppress the invariance to reflection of input data in x direction.

Both operators f_0^{x+y} and f_0^{xy} suppress the invariance to reflection in both direction. Due to their simplicity is operator f_0^{xy} preferable to operator f_0^{x+y}.

In this model it's preferable to use NT and modified NT with functions $max(a, b)$ and $min(a, b)$. One of the reasons is that in next level of the model will be necessary to know maximal components of characteristics of the symbol - vectors $p_i(\theta)$, i= 1, 2, ... and transform NT reorganize this components of vectors, so the maximal component will be the first in a row. Network doesn't have to be more complicated by functions, which search for maximal component of the vector.

F. Classification

Complete classification within the process of symbol recognition we describe as identification of unknown input symbol, which means to find class of symbols based on selected comparing criteria, which characteristics are mostly matching the characteristics on unknown symbol.

It's possible to divide the methods into syntactic and mathematical. Mathematical methods of classification can be then divided into deterministic methods and statistic.

Some of the simplest classifiers based on statistic methods are classifiers that are using Euclidian distance:

1-dimensional: $$d_E^{(1)} = \sqrt{\sum_{i=1}^{N} (x(i) - \tilde{x}(i))^2}, \quad (16)$$

2-dimensional: $$d_E^{(2)} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M} (x(i,j) - \tilde{x}(i,j))^2} \quad (17)$$

where N, M are dimensions of 1-dimensional $\overline{\mathbf{X}}$ or 2-dimensional $[\mathbf{X}]$ of input signals and $\tilde{\overline{\mathbf{X}}}$ or if you like $[\tilde{\mathbf{X}}]$ is 1-dimensional or 2-dimensional reference signal of classifier.

Due to the character of extracted features - vectors p_i, i= 1, 2,… N, the 1-dimensional Euclidian distance was used for classification of input of the previous levels $d_E^{(1)}$ (16).

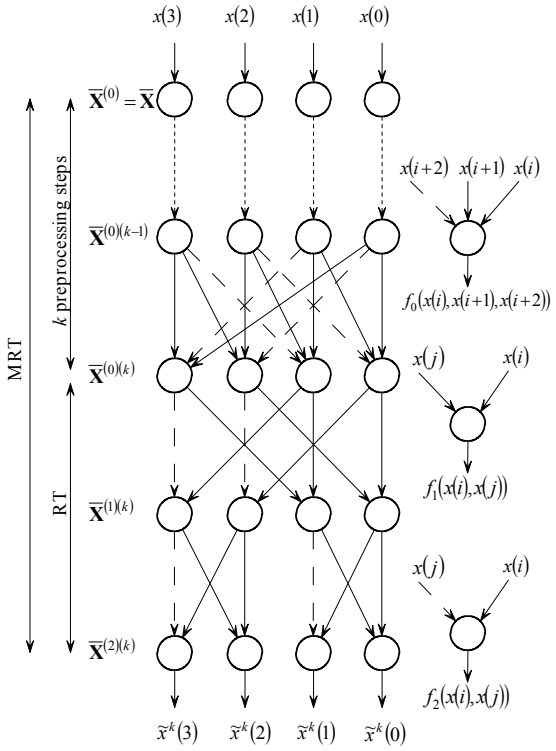


Figure 2. Signal graph of MRT calculation (for N=4)

III. EXPERIMENTAL AND SIMULATION RESULTS

There were tested several properties of the system for invariant symbol recognition using program tools, created in Matlab. In the experiments the system was tested if it can recognize the most important prohibitory, priority and warning

traffic signs, which were represented by several binary images with the size of 347x347 pixels (Fig. 3). The goal of the experiments was also test the system's ability to recognize the 15 traffic signs under hardened conditions such as changed scale and shifted positions in the image. Each of the traffic signs was changed in scale from 0.25 to 1.26 of the original. In addition, the position of the scaled (scale 0.25) traffic signs in the image was significantly shifted (multiple of the size of recognized object horizontally and vertically). The complete number of the images with shifted and scaled images resulted in set of 91 tested images.

The traffic sign recognition was tested with the RT and MRT. The successfulness of traffic sign recognition with RT or MRT was tested with the number of four moments. However the MRT as the more sophisticated algorithm which has the option of preprocessing was set to different number of pre processing. The number of preprocessing was set from 1 to 5. In this way we were able to examine the successfulness of the traffic sign recognition with MRT under different settings.

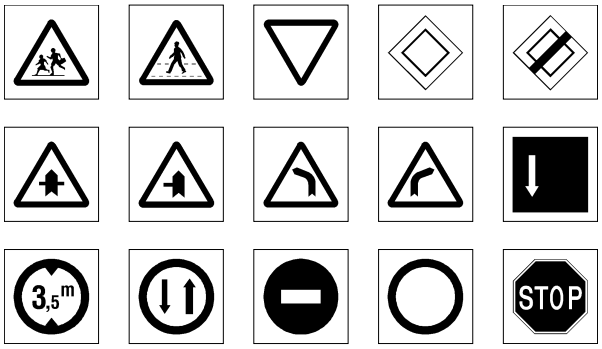


Figure 3. Images with traffic signs

The results of the experiments were as follows. The recognition with the RT transform was able to recognize 85% of the overall set (91 images) of traffic signs. The results of the recognition procedure with MRT are different in the manner of how many steps of preprocessing were selected. The influence of different settings of the number of preprocessing steps on the successfulness of the recognition with MRT is shown in the Table II.

TABLE II. SUCCESFULNESS OF MRT

Number of preprocessing steps	1	2	3	4	5
Successfulness	82%	87	91%	96%	78%

IV. CONCLUSION AND FUTURE WORK

From the results of the presented experiments it is significant, that the RT and MRT have comparable successfulness, if the number of pre processing steps of MRT is set to 1 or 2. With the increasing number of preprocessing steps the reliability of the MRT algorithm is increasing. However, if the number of the preprocessing steps is high, the MRT cannot work properly which results in a lower successfulness of the traffic sign recognition. The optimal number of the preprocessing steps appears to be 4, where the MRT was able to recognize the 96% of the tested traffic signs.

Also the experiments confirmed our notion, that the algorithm whether RT or MRT were not able to recognize the traffic signs which scale was 0.25 of the original. Such small images with traffic signs had relatively low quantity of information and hence could not be recognized.

In the future work we want to focus on the traffic sign recognition on the gray scale images.

ACKNOWLEDGMENT

We support research activities in Slovakia / Project is co-financed from EU funds. This paper was developed within the Project "Centrum excelentnosti integrovaného výskumu a využitia progresívnych materiálov a technológií v oblasti automobilovej elektroniky", ITMS 26220120055

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[2] YOU, S. D. - FORD, G. E. Connectionist Model for Object Recognition, Applications of Artifical Neural Networks III, Vol. 1709, 1992, 200-207

[3] [2] YOU, S. D. - FORD, G. E. Object Recognition Based On Projection, Proceedings of 1993 International Joint Conference on Neural Networks, 1993, IV-31 - IV-36

[4] [3] YOU, S. D. - FORD, G. E. Network model for Invariant Object Recognition and Rotation Angle Estimation, Proceedings of 1993 International Joint Conference on Neural Networks, 1993,

[5] [4] SHAMS, S. Translation-, Rotation-, Scale-, and Distortion-Invariant Object Recognition Through Self-Organization, International Journal of Neural Systems, Vol. 8, No. 3, June 1997,173-179

[6] [5] VERMA, B. Recognition of Rotating Images Using an Automatic Feature Extraction Technique and Neural Networks, International Journal of Neural Systems, Vol. 8, No. 3, June 1997, 201-207

[7] [6] TURÁN, J. Fast translation invariant transform and their applications. Košice : Elfa, 1999. 158 s. ISBN 80-88964-19-9.

[8] [7] TURÁN, J. Acoustic Object Recognition with Use of Rapid Transform, First Japanese-Czechoslovak Joint Seminar on Aplied Electromagnetics, June 17-19, 1992, 87-91.

[9] TURÁN, J. Recognition of Printed Berber Characters Using Modified Rapid Transform, J. Of Communications, Vol. XLV, 1994, 24-27.

Experimental Evaluation of Regular Events Occurrence in Continuous-time Markov Models

Vaclav Vais
Department of Computer Science and Engineering
University of West Bohemia
Plzen, Czech Republic
vais@fav.zcu.cz

Stanislav Racek
Department of Computer Science and Engineering
University of West Bohemia
Plzen, Czech Republic
stracek@kiv.zcu.cz

Abstract—**Continuous-time Markov process is widely used abstract tool to construct high-level models of complex computer systems in order to evaluate either performance or reliability parameters of the system. Utilization of the continuous-time Markov process is based on an assumption of exponential distribution of the time between random events influencing behavior of the modeled system. Another probability distribution of this time needs an adaptation of the original model. This article uses a representative example to evaluate precision of the modeled system parameters when the exponential distribution of burning time of events is replaced with another probability distribution, including regular distribution of events.**

Keywords; Markov model, influence of probability distribution, experimental evaluation

I. INTRODUCTION

A powerful tool for analyzing some probability based systems and or problems from the area of computer science is the probabilistic model based on the mathematical theory of the (stochastic) *Markov processes*. For a thorough review of the basic theory, please consult e.g. [2]. From a computer researcher point of view, Markov process is a kind of finite automata, where a transition between two states is caused by random event, i.e. the time duration of every state is a random variable. When the state is reached, the transition is *fired.* Every transition (an edge in the graph describing the process) has assigned a value of *transition rate*. This value can be interpreted in two ways: (i) it is conditional (transition is fired) frequency of the (subsequent) transitions, and (ii) it is the parameter of the exponential probability distribution of the time interval between the transition firing and transition occuring (denoted here as time interval of the *transition burning*). The model (i.e. its state-transitions graph) can be easily transformed into a set of linear differential equations from which time dependent state probabilities $p_0(t)$, $p_1(t)$, … can be computed using conventional methods.

Markov models are used in two basic categories. First category contains models with one or more *absorbing states*, i.e. states without an output edge. It is apparent that these models have "limited time of life". Time dependent probabilities of model states are computed directly from the corresponding set of differential equations, then the target parameters can be determined, usually as a linear combination of some state probabilities. Markov models from the second category have "infinite life" (i.e. no absorbing states) and here the asymptotic probabilities (i.e time independent limits $p_0=p_0(\infty)$, $p_1=p_1(\infty)$, ...) of model states can be computed from a set of linear algebraic equations. Subsequently significant parameters can be determined using known values of the model states asymptotic probabilities. The analyzed case used in this article falls into the second mentioned category. Description of Markov models utilization in the area of computer science can be found e.g. in [1], [3], [4].

Utilization of Markov processes is limited by the assumption of exponential probability distribution of the duration of any transition burning. Exponential distribution is quite "irregular", i.e. its standard deviation has the same value as its mean value (both are $1/\lambda$, where λ is (the single) parameter of the distribution). In some applications (see e.g. [7]), the Markov model is constructed and utilized even when the modeled system time behavior is influenced by more regular events, e.g. by built-in tests with a quite regular period. This article has the aim to use a representative example to evaluate numerically a deviation which occurs when we use Markov model and the assumption of exponential distribution of events burning time is not quite valid.

II. SYSTEM TO BE EVALUATED

For the analysis we have chosen classical model of cooperating parallel processes. The assumed computational environment can be e.g. symmetrical multiprocessor system consisting of n processors and a shared memory. We have n computational processes, every process has its own processor. The processes cooperates using one *critical section* containg all the shared data of processes. All processes have the same program describing their cyclic behavior: local computation without any interaction with another process, then computation within the critical section, etc. The computation inside critical section needs to be locked, i.e. only one process in one time can be in this part of (the shared) program. All

processes have the same time behavior with the following parameters:

- λ … mean conditional frequency of a process local computation, i.e. reversed value of the mean time of the local computation,
- μ … mean conditional frequency of any computation inside critical section, i.e. reversed value of the mean time of this computation.

The described example can serve as a model of parallel computation based on utilization of data parallelism – a process works on separate (local) piece of a large set of data, then updates (global) result of computation. This activity is performed periodically until all the set of data is exhausted. Time intervals of the processes behavior needs then to be taken as random variables due to different values of data processed within different cycles of activity. When

$$(n-1) / \mu < 1/\lambda$$

then the ideal (linear) speedup of parallel computation $s_{max} = n$ can be reached assuming deterministic time behavior. When the processes have a random time behavior, their conflicts (i.e necessary synchronization at the input of critical section) decreases the reachable value of the speedup. Keeping the above stated condition, maximum frequency (i.e. frequency without conflicts) of every process computation is as follows:

$$f_{max} = 1/ (1/\lambda + 1/\mu)$$

Due to the conflicts, the real frequency of computation f is decreased compared to f_{max}, so we can define a speedup degradation coefficient d as the ratio:

$$d = f_{max} / f$$

Corrected value of the speedup can be then expressed as

$$s = s_{max} /d = n/d .$$

When the time intervals of local computation and the time intervals within critical section have the exponential distribution (i.e. they are quite irregular), then the analytic solution for the degradation coefficient d can be found (see the next section). Variables λ and μ are then taken as the parameters of the corresponding exponential probability distribution and the reversed values $1/\lambda$ and $1/\mu$ serve as the mean values of the corresponding distribution. But an assumption of exponential distribution (i.e. total irregularity) of the processing time intervals can be questionable, especially in the case of time spent inside the critical section. In the analyzed example (model of parallel computing with data parallelism utilization) this time corresponds a global result update, which operation can be quite regular. That is why in section IV an influence of increased regularity of the time spent inside the critical section will be taken into account, still using (modified) Markov model. In section V an influence of combined regularity of the both processing times will be computed by means of discrete-time simulation model.

The basic Markov model (see below, Sec.III) is general enough and it can be used as an abstract model of many systems or problems from the area of the applied computer science. For example we can use it for a closed queuing network with one server (here μ is the serving rate) and n clients which are non-stop generating (with the rate λ) their requests to be processed at the server. Here the degradation coefficient d reflects time lost with clients unproductive waiting for the server to start their request. Another example is from the area of fault-tolerant systems. Highly available information system uses n (identical) servers. The fault rate of a server is then λ . Rate of repairs is μ, these are performed in a sequence (one repairman assumption). In both given examples, the assumption that λ represents parameter of exponential probability distribution is acceptable (irregular corresponding times), but similar assumption for μ is questionable (more regular time of single services or more regular time of single repairs).

III. ANALYTICAL SOLUTION

The Markov model of an example described above can be represented by a simple state diagram.

Figure 1: The basic Markov model

This system can be represented by matrix equation

$$\mathbf{A}\,\mathbf{p} = \mathbf{0}$$

where $\mathbf{p} = (p_0 , p_1, ..., p_{n-1} , p_n)^T$ is a column vector of asymptotic probabilities and $\mathbf{A}$ is a matrix coefficient of the system

$$\mathbf{A} = \begin{bmatrix} n.\lambda & -\mu & 0 & . & . & 0 & 0 \\ -n.\lambda & (n-1).\lambda+\mu & -\mu & . & . & 0 & 0 \\ 0 & -(n-1).\lambda & (n-2).\lambda+\mu & . & . & 0 & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & . & . & . & -\mu & 0 \\ 0 & 0 & 0 & . & . & \lambda+\mu & -\mu \\ 0 & 0 & 0 & . & . & -\lambda & \mu \end{bmatrix}$$

Generally any system of linear equations representing Markov process by this way is linearly dependent. The rank of matrix

A is n (number of states minus 1). This degradation will be eliminated by replacing any equation by equation:

$$p_0 + p_1 + \ldots\ldots\ldots + p_{n-1} + p_n = 1$$

(system will always be in some state with probability 1).

The form of matrix **A** clear the way for deriving of analytical solution of vector **p**. Suppose that a value of p_0 is known. Due to the first row of matrix **A** p_1 can be expressed straightly in terms of p_0, due to the second row p_2 can be expressed straightly in terms of p_0 and p_1, etc. After a sequence of algebraic transformations all the probabilities p_i can be expressed in terms of p_0 (in this special case, not in general):

$$p_i = p_0 \cdot \left[\prod_{j=0}^{i-1} (n-j) \right] \left(\frac{\lambda}{\mu} \right)^n$$

Then it follows:

$$p_0 = \frac{1}{1 + \sum_{i=1}^{n} \left\{ \left[\prod_{j=0}^{i-1} (n-j) \right] \left(\frac{\lambda}{\mu} \right)^n \right\}}$$

The real frequency of computation (i.e. frequency with conflicts) is :

$$f = \frac{\mu.(1-p_0)}{n}$$

The table I. shows numerical values of speedup degradation coefficient d based on the analytical model presented above. These results were computed for a representative set of parameters. The meaning of parameters is explained in the previous text. For example the ratio $\lambda / \mu = 0.1$ means ten times longer local computation (in average) compared to the average time of the critical section utilization. Value $d = 1.0424$ for $n = 5$ processes means about 4% longer computation due to the influence of conflicts when accessing the critical section.

TABLE I.

λ/μ	n	2	3	5	10
0,01		1,0001	1,0002	1,0004	1,0010
0,1		1,0083	1,0179	1,0424	1,1575
0,2		1,0278	1,0631	1,1653	1,6979
0,5		1,1111	1,2667	1,7302	3,3335

IV. NUMERICAL SOLUTION

In this section the time intervals inside the critical section will be observed as a serial connection of k stages, each of them with the exponential distribution. The mean conditional frequency of any stage is $k.\mu$. Therefore the aggregate time within the critical section has Erlang-k distribution in this case. The method of stages is discussed in [5]. In general, the state with non-exponential distribution can be split into some serial-parallel cluster of two or more exponentially distributed stages.

The following state diagram represents Markov model where computation inside critical section is divided into k stages with identical mean times of all stages of computation.

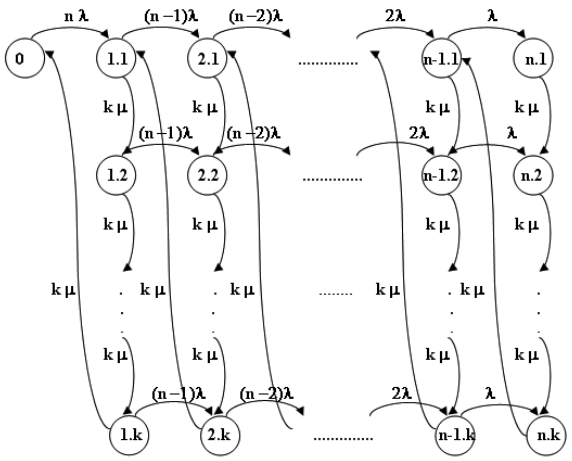


Figure 2: The extended Markov model

Asymptotic probabilities of model states can be computed from a system of $n.k + 1$ linear algebraic equations. We will illustrate it using case for $n = 3$ and $k = 2$.

$$\begin{bmatrix} 3\lambda & -2\mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\lambda+2\mu & -2\mu & 0 & 0 & 0 & 0 \\ -3\lambda & 0 & 2\lambda+2\mu & -2\mu & 0 & 0 & 0 \\ 0 & -2\lambda & 0 & \lambda+2\mu & -2\mu & 0 & 0 \\ 0 & 0 & -2\lambda & 0 & \lambda+2\mu & -2\mu & 0 \\ 0 & 0 & 0 & -\lambda & 0 & 2\mu & -2\mu \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

This matrix equation is not suitable for deriving of the analytical solution. In general, probabilities p_i cannot be simply expressed in terms of p_0. Therefore numerical solution of this system seems to be the best way to obtain vector of asymptotic probabilities $\boldsymbol{p}$. We used the standard MS Excel software to automatically construct the matrix coefficient **A**, to solve the system of linear equations, and to compute the real frequency of computation f, and a speedup degradation coefficient d.

The next table shows the numerical values of speedup degradation coefficient d based on the numerical model

presented above. This results were computed for the same set of parameters as in section III. The lower row presents the relative deviation between the exact analytical solution (i.e. for exponential distribution of the intervals inside the critical section) and the numerical solution for $k = 10$ stages with exponential distribution.

TABLE II.

λ/μ	n	2	3	5	10
0,01	results	1,0001	1,0001	1,0002	1,0005
	deviation	0,00%	0,01%	0,02%	0,04%
0,1	results	1,0048	1,0105	1,0258	1,1134
	deviation	0,34%	0,72%	1,59%	3,81%
0,2	results	1,0170	1,0400	1,1172	1,6725
	deviation	1,05%	2,17%	4,13%	1,50%
0,5	results	1,0759	1,2045	1,6844	3,3333
	deviation	3,17%	4,91%	2,64%	0,00%

V. SIMULATION BASED SOLUTION

The last part of our analysis is aimed to evaluate combined influence of increased regularity of both probability distributions – distribution of the process local activity computation time as well as distribution of the time spent inside the critical section. We can still use the method of stages explained in the previous section. But the resultant Markov model is complex enough and its complexity growth exponentially both with number of processes and number of assumed stages of both activities. So for this case we decided to use a simulation model. The used model is discrete-time and Monte Carlo based, i.e. it uses random numbers to determine single values of duration of modeled processes activities. As the implementation tool we used C-Sim library [6]. The simulation model can be easily verified when we use it for the cases described above in sections III and IV and when we compare the results. If we let to run the simulation program for 10^6 cycles of modeled processes then the relative error of the computed d is about 10^{-3}.

TABLE III.

λ/μ	n	2	3	5	10
0,01	Results	0,9996	0,9998	1,0000	0,9998
	Deviation	0,05%	0,04%	0,04%	0,12%
0,1	Results	1,0041	1,0095	1,0233	1,0935
	Deviation	0,41%	0,82%	1,83%	5,53%
0,2	Results	1,0146	1,0337	1,0932	1,6655
	Deviation	1,28%	2,76%	6,19%	1,91%
0,5	Results	1,0539	1,1525	1,6675	3,3306
	Deviation	5,15%	9,01%	3,62%	0,09%

The table III was computed for the same set of parameters as previous two tables. It shows results obtained when both time intervals were divided into $k = 10$ stages, i.e. the modeled probability distribution was the Erlang`s distribution of the k-th degree. For this case the coefficient of variance C (as a measure of regularity) for both distributions is $C = 1/\sqrt{k} = 0.32$. It approximately corresponds to the Gaussian distribution with the standard deviation about one third of its mean value.

When comparing the results with the previous table(s), we can see the expected influence of the increased regularity – the computed d has better (i.e. decreased) values. When we use simulation model for quite regular values for both time intervals of activity, we obtain the expected result (processes are fully synchronized) $d = 1.0$ with a sufficient precision.

VI. CONCLUSION AND FUTURE WORK

This article uses a representative example to evaluate precision of the system parameters computed using Markov model, when the assumed exponential distribution of burning time of events is replaced with another probability distribution. The chosen example is from the area of parallel processing, but the results can be generalized into another parts of computer science, e.g. queuing networks or fault-tolerant systems. The results show, that for an integral parameter like the evaluated degradation coefficient, the deviation of result created when we replace exponential distribution of events burning time with another distribution is not too large. Within our analysis this deviation did not exceed 20%. In fact, the evaluated degradation coefficient d is combined from probabilities of many states of the used Markov model where deviations in evaluated probabilities of single states can eliminate each other. It is possible to expect, that probability values (or time functions) of some (chosen) states of the model can be influenced by the change of the events burning time probability distribution much more, but it is the matter of our future work.

REFERENCES

[1] J.Hlavička et al.: Číslicové systémy odolné proti poruchám, In: Vydavatelství ČVUT, Praha, Czech rep. (1997), p.330.

[2] P. Mandel: Pravděpodobnostní dynamické metody, In: Academia, Praha (1995), p.181.

[3] D.P.. Siewiorek – R.S. Swartz: The Theory and Practice of Reliable System Design, In: Digital Press, Bedford, USA-MA (1996), p.772.

[4] K.S. Trivedi: Probability and Statistics with Reliability, Queuing ad Computer Science Applications, In: Prentice Hall, USA, (1998), p. 623.

[5] R. Billinton – R.N. Allan: Reliability Evaluation of Engineering Systems, In: Plenum Press, New York, USA (1983), p.349.

[6] http://www.c-sim.zcu.cz

[7] R.Dobias – J.Konarski – H.Kubatova: Dependability Evaluation of Real Railway Interlocking Device, In: Proceedings of IEEE CS, 11th Euromicro Conference on Digital Systems Design, (2008), pp. 228 – 233.

Increasing Performance in Distributed File Systems

Pavel Bžoch
Department of Computer Science and Engineering
University of West Bohemia
Pilsen, Czech Republic
pbzoch@kiv.zcu.cz

Jiří Šafařík
Department of Computer Science and Engineering
University of West Bohemia
Pilsen, Czech Republic
safarikj@kiv.zcu.cz

Abstract — **Need of storing huge amounts of data has grown over the past years. Whether data are of multimedia types (e.g. images, audio, or video) or are produced by scientific computation, they should be stored for future reuse or for sharing among users. Users also need their data as quick as possible. Data files can be stored on a local file system or on a distributed file system. Local file system provides the data quickly but does not have enough capacity for storing a huge amount of the data. On the other hand, a distributed file system provides many advantages such as reliability, scalability, security, capacity, etc. In the paper, traditional DFS like AFS, NFS and SMB will be explored. These DFS were chosen because of their frequent usage. Next, new trends in these systems with a focus on increasing performance will be discussed. These include the organization of data and metadata storage, usage of caching, and design of replication algorithms.**

Keywords-distributed file systems; file replication; caching; performance

I. INTRODUCTION

Distributed systems (DS). Modern scientific computations require powerful hardware. One way of getting results in scientific work faster is purchasing new hardware over and over again. Buying powerful hardware is, however, not a cheap solution. Another way is using distributed systems, where the need for performance is spread over several computers. In distributed systems, several computers are connected together usually by LAN. In the client's view, all these computers act together as one computer. This concept brings many advantages. Better performance can be achieved by adding new computers to the existing system. If any of the computers crash, the system is still available. Using DS brings several problems too. In the distributed systems, we have to solve synchronization among computers, data consistency, fault tolerance, etc. There are many algorithms which solve these problems. Some of them are described in [1].

Distributed file systems (DFS). Distributed file systems are a part of distributed systems. DFS do not directly serve to data processing. They allow users to store and share data. They also allow users to work with these data as simply as if the data were stored on the user's own computer. Compared to a traditional client-server solution, where the data are stored on one server, important or frequently required data in DFS can be stored on several nodes (node means a computer operating in a DFS). This is called *replication*.

The data in a DFS are then more protected from a node failure. If one or more nodes fail, other nodes are able to provide all functionality. This property is also known as *availability.* The data replication also increases system performance – a client can download a file from the node which is the most available one at a given moment.

Files can also be transparently moved among nodes. This is typically invoked by an administrator and it is done for improving load-balancing among nodes. The users should be unaware of where the services are located and also the transferring from a local machine to a remote one should also be transparent [2]. In DFS this property is known as *transparency.*

If the capacity of the nodes is not enough for storing files, new nodes can be added to the existing DFS to increase DFS capacity. This property is also known as *scalability*.

A client usually communicates with the DFS using LAN, which is not a secure environment. Clients must prove their identity, which can be done by authenticating themselves to an authentication entity in the system. The data which flow between the client and the node must be resistant against attackers. This property is known as *security*.

II. SUMMARY OF TRADITIONAL DFS

This section will describe traditional DFS. There are many DFS, some of which are commercial (like AFS), and others free (like OpenAFS, NFS, Coda and Samba). This section will describe traditional DFS like NFS4, OpenAFS, Coda and SMB. Many of the new DFS extend or are based on these traditional DFS.

A. OpenAFS

AFS (Andrew File System) was originally created at Carnegie Mellon University; later it was a commercial product supported by IBM. Now it is being developed under a public license. AFS has a uniform directory structure on every node. The root directory is /afs. This directory contains other directories which correspond to the *cells.* Cells usually represent several servers which are administratively and logically connected. One cell consists of one or more *volumes*. One volume represents a directory sub-tree, which usually belongs to one user. These volumes can be located on any AFS server. Volumes can be also moved from one

AFS server to another. Moving volumes does not influence the directory tree. Information about the whole system is stored in a special database server [3]. AFS supports client-side caching. Cached files can be stored on a local hard disk or in a local memory. Frequently used files are permanently stored in this cache. AFS does not provide access rights for each file stored in the system, but it provides directory rights. Each file inherits access rights from the directory where the file is located. For achieving better performance of read-only files, *replica servers* can be used. When the other servers are overloaded, the replica server provides files to the clients instead of these servers. AFS is a very stable and robust system and it is often used at universities. AFS uses Kerberos [4] as an authentication and authorization mechanism. More information about AFS can be found in [5].

B. NFS4

NFS (Network File System) is an internet protocol which was originally created by Sun Microsystems in 1985, and was made for mounting disk partitions located on remote computers. NFS is based on RPC (Remote Procedure Call) and is supported in almost all operating systems. The NFS client and server are a part of the Linux kernel. The Kerberos system is used for user identification. System performance is increased by using a local client cache. NFS has two main elements: a client and a data server. NFS can be extended into pNFS (parallel NFS), which contains one more server called metadata server. The metadata server can connect a file system from any data server to a virtual file system. It also provides information about the file location to the clients. When clients write file content, they must also ensure file updating on all servers where the file is located. NFS communicates on one port since version 4 (previous versions used more ports), so it is easy to set up a firewall for using NFS4 [3]. More information about NFS can be found in [6].

C. Coda

Coda was developed at Carnegie Mellon University in 1990. It is based on the AFS idea and is implemented as a client and several servers. This system was mainly designed to achieve high availability. The client uses a local cache. Coda supports off-line working, which means that cached files are available even after disconnection from the server. While the client is disconnected from the server, all changes made to files are stored in a local cache. After reconnecting to the server, all these changes are propagated to the server. If any collision occurs, the user has to solve it manually. Coda uses Kerberos as an authentication and authorization mechanism. Servers provide file replication for achieving availability and safety. Coda uses RPC2 for communication. Servers store information about files which are in the client's cache [3]. When one of the cached files is updated, the server marks this file as non-valid.

D. SMB

SMB was developed in 1985 by IBM as a protocol for sharing files and printers. In 1998 Microsoft developed a new version of SMB called Common Internet File System (CIFS), which uses TCP/IP for communication. SMB has been ported to other operating systems where the SMB is called *samba*. This system is stable, wide-spread and comfortable. SMB can use Kerberos for authentication and authorization of users. It does not use local client-side caching and uses the operating system's file access rights.

III. Data and Metadata Storage Organization

This section will describe modern trends in DFS with a focus on data storage and metadata storage. Data storage is used for storing file content. Metadata storage usually stores file attributes and links to the file content in the data storage.

A. Data storage

Data storage is used for storing file content. When users want to upload a file to the DFS, they send the entire file to the server. On the server side, this file is split into two parts: file content and file metadata. File content is then stored in a data storage node. The whole uploading process is depicted in Figure 1.

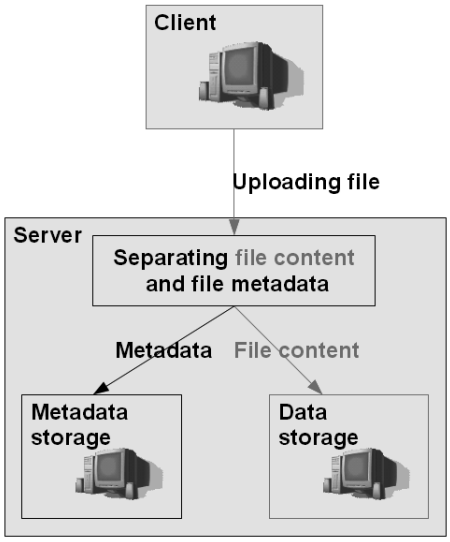


Figure 1. File uploading process (file content)

For the data storage, local hard disks with their own file systems are usually used. On nodes with OS Microsoft Windows, NTFS is commonly used. In UNIX-like systems, several different file systems exist. Not all of these systems are suitable for all types of files. According to the tests in [7], the RaiserFS is more efficient in storing and accessing small files, but it has a long mount time and is less reliable than EXT2/3. XFS and JFS have good throughput, but they are not efficient in file creation. EXT2/3 has severe file fragmentation, degrading performance significantly in an aged file system [7]. The decision on which file system will be used for data storage is an important part of DFS design.

Another method of storing file content is a designing new data organization of a hard disk. This concept is used when the existing data organization (file system) of a hard disk is not suitable for files which will be stored there. New hard

disk organization is shown in [8]. In this proposal, every single *inode* can be locked without increasing the total amount of locks. Distribution of the inodes in this proposal also makes the system more expandable because the distribution makes it possible for the number of inodes to increase or decrease dynamically.

Writing and reading file content or creating a new file is a slow operation. I/O operations are the bottleneck in achieving better performance in DFS. Uploading and storing files on a server have several steps. These steps must be done chronologically to ensure data consistency. The steps are: sending a file to the server → splitting file content and file metadata → creating a new metadata record → creating a new file and storing file content → connecting metadata with the file handle. Both, creating the metadata record and storing content, are slow operations. According to [9] and [10], these slow operations can be accelerated. Paper [9] presents increasing performance by making changes in an upload protocol. Paper [10] presents increasing performance by reducing the number of messages which are sent during the upload process. Both [9] and [10] demand cooperation between the file storage nodes and the metadata storage nodes.

Papers [7], [8], [9] and [10] assume that the whole file is stored in one node. Another way of storing files is splitting a file content into file fragments and storing these fragments on the client side. This proposal does not work on a client-server model, but works in P2P networks. Links to the file fragments are stored in a distributed hash table. This system also provides file replication. The entire system is described in [11]. The P2P system architecture is depicted in Figure 6.

B. Metadata storage

Metadata are a specific type of data which give us information about a certain item's content. In DFS, metadata are used for providing information about files which are stored in the data storage. This information is usually called file attributes. These attributes are the date and time of file creation, the date and time of the last modification, the file size, the file owner, the file access rights, etc.

Metadata storage also provides information about a directory structure. All this information is created during the upload process (see Figure 2). Each record must also have a link to the data storage. Metadata storage must provide functions for getting and storing file metadata, file searching, moving files within directories, deleting files and creating files. Additionally, metadata storage can provide locks for ensuring consistency during the file access. There usually are two types of locks: a shared lock for file reading and an exclusive lock for writing or updating file content. Metadata are usually stored in a database or in tree.

Database records are used, e.g., in the AFS. While using the database, all metadata operations are represented by a database query. Trees are used in [12] and [13]. Entire tree is usually stored in RAM. Tree is used for maintaining namespace information. Adding a new file metadata record is simply adding new node to the tree. Node corresponding to the file must also have a link to the file content in the data storage. When the client requests a file, the algorithm described in [12] returns links to all data storages where file replicas are stored. The algorithm described in [13] returns a link to the data storage which is the closest to the client. The metadata node keeps a list of available data nodes by receiving heartbeat messages from these nodes.

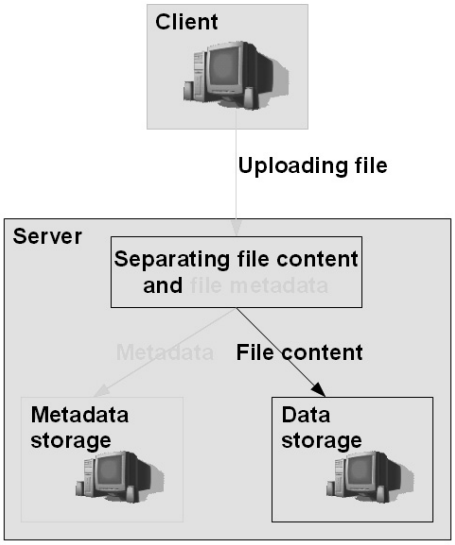


Figure 2. File uploading process (metadata)

Another way of storing metadata involves using a log-structured merge tree (LSM). LSM tree is multi-version data structures composed of several in-memory trees and an on-disk index [14]. Paper [14] describes the application of LSM to metadata storage for distributed file systems. The database consists of a database log, an on-disk search tree and a checkpoint mechanism.

IV. CACHING AND REPLICATION ALGORITHMS

The previous section describes how the choice of data and metadata storage can influence DFS performance. This section will describe caching and replication algorithms which can also increase DFS performance.

A. Caching

A cache in the computer system is a component which stores data that can be potentially used in the future. When the cached data are requested, the response time is shorter than when the data are not in a cache and must be downloaded. There are several caching policies which try to predict future requests. Caching policies are used to mark the entity which can be removed from the cache when a new entity comes to the cache. Most of these algorithms are based on statistics made from previous data requests. Most of caching policies are described in [15]. The most effective replacement policy is OPT, but OPT cannot be implemented in practice since that would require the ability to look into the future. According to [15], the most effective replacement policy is LRU.

Many papers describe which of the cache policies is the most effective for use in a DFS. There are also some modifications of these policies for increasing cache-hit ratio.

Paper [16] extends existing LRU and LFU policies with a Size and Threshold policy. An LRU or LFU policy makes an ordered list of files which can be removed from the cache according to the LRU or LFU algorithm. LRU or LFU with Size means that the size of the file which will be removed must be greater or equal to the size of the new file. LRU or LFU with Threshold means that the size of the file which will be removed must be greater or equal to the threshold value. The most effective policy in [16] is, again, LRU with no extension.

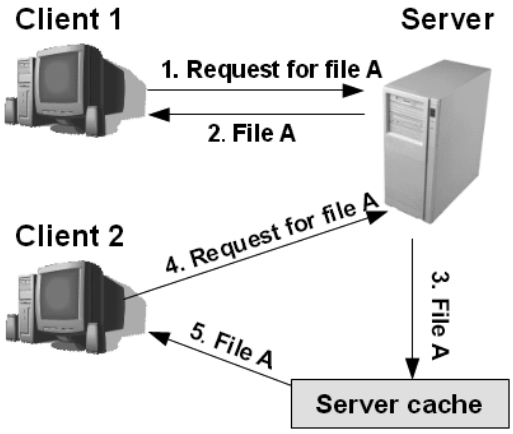


Figure 3. Server-side caching

A cache can used be on server side, on client side or on both side of the system. The server stores in the cache the data which are frequently requested by clients (see Figure 3). At a server side, cache can be stored in-memory or on separate machine. The client stores in the cache the data which may be requested again in the future (see Figure 4). Client cache is usually stored in-memory. If there are both caches in the system, the server cache-miss ratio increases. Clients often request files in their own cache, so they do not need the server to get the file. On the other hand, the server gets requests on different files, so the server-side cache is useless. This case is described in [17].

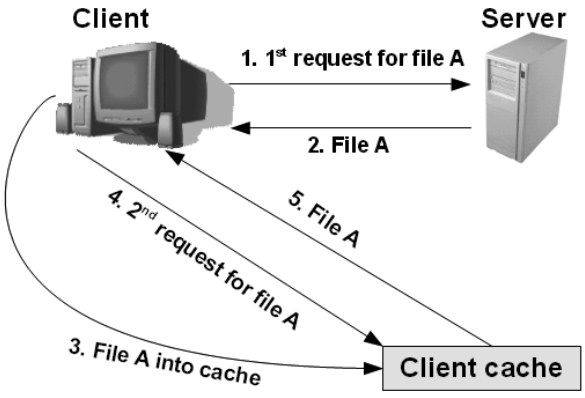


Figure 4. Client-side caching

Paper [18] presents a decentralized collective caching architecture. In this paper, caches on the client side are shared among clients. When a client downloads a file, this file is then stored in the client's cache. The server stores a list of clients to each file. This list also contains the client network address. When another client wants to access this file, the server returns this list. The client can then download the file from one of the listed clients. The server then adds this new client to the list. This proposal decreases server work-load, but it requires cooperation among clients. Paper [18] also describes consistency semantics. When new file content is uploaded to the server, the server invalidates file copies in the clients' caches.

Another way to reduce server workload and network bandwidth is by using proxy caching. Proxy caching in the DFS is introduced in [19]. A proxy cache stores files which are requested by clients. The whole cache is stored in a proxy server. The proxy server in this paper is on a local network. This paper also assumes that the connection to the server is slow. All file requests to the remote server pass through this proxy server. If the requested file is in the proxy cache, the proxy server returns this file and there is no need to communicate with the remote server.

B. Replication

Replication in a DFS is a process whereby the original file is copied to other servers in the DFS. The original file is usually called *the primary replica* or *master replica;* other copies are called *replicas*. A replication algorithm (or strategy) describes which data will be replicated, as well as how, when, and where the replicated file should take place [20]. Replication can be used for achieving better performance, availability or fault tolerance. All these three requirements use slightly different algorithms for choosing the file and the place for the replication.

We will focus on replication for achieving better performance. In this replication, choosing the file and the place for replication is very important. The file for replication should be read very often and should not be modified very often. Writing or updating a replicated file is an expensive operation. Choosing a place for a file replica is also very important. The server which is chosen to store the replica should not be over-loaded and should have good network connectivity.

Replication can be done statically (administratively) or dynamically. In a static replication system, the administrator marks storage where the primary replica is placed. The administrator then defines the number of replicas and the replicas' placement. In this case, the administrator predicts which files will probably be the most used in the future.

Another method of file replication is dynamic replication (see Figure 5). Dynamic file replication is described in [21] and [22]. Both of these papers represent dynamic replication based on statistical information. The statistical information can be the availability of servers, the workload of servers working in DFS, the average response time, or the number of requests for the file. Based on this information, the files for replication and the server are chosen.

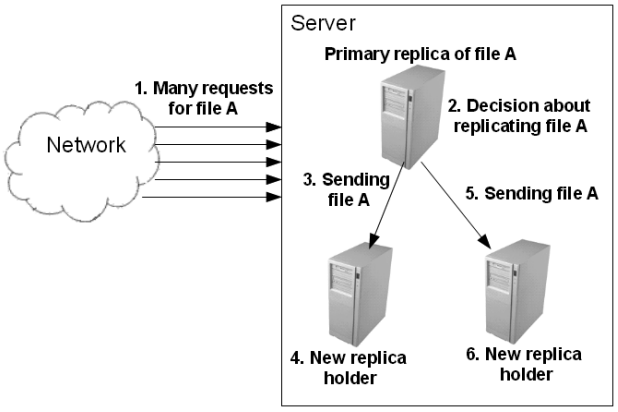


Figure 5. Dynamic file replication

Another dynamic replication is described in [11]. This paper uses P2P architecture where fragments of files are stored on several peers. The system architecture is depicted in Figure 6. The peer closest to the file fragment ID is responsible for that fragment and has to check periodically if enough replicas are available [11]. If there are not enough replicas in the system, the peer can replicate fragments.

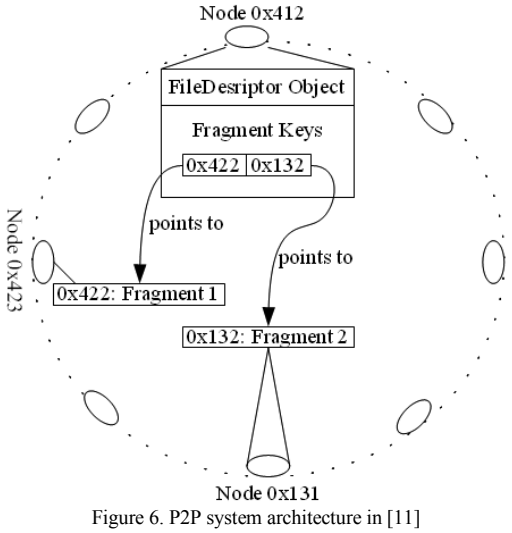


Figure 6. P2P system architecture in [11]

File replication for increasing performance is also used in other DFS. CloudStore [23] typically uses 3-way file replication (files are typically replicated to three nodes). If there is a need for replication (e.g. node outage), a metadata server can replicate a file chunk to another node. This conception of file replication is derived from the Google File System [24]. GlusterFS [25] uses three file replication options. The first option is a file distribution over mirrors. This means that each storage server is replicated to another storage server. Other ways of storing files are file distribution to one node or file stripping over nodes. These two solutions are less reliable.

V. SUMMARY

This paper describes methods which can be used for increasing a distributed file system's performance. System performance can be increased by choosing a suitable file system. If there is no suitable file system, a new one can be developed.

Another way of increasing performance is by accelerating I/O operations, which are the bottleneck of system performance. To achieve better performance a metadata storage scheme is also important. Metadata operations make up over half of the workload of the DFS. As metadata storage, a database or a special data structure (like B-trees, LSM trees) can be used. Databases are usually used in the traditional DFS (such as AFS) and trees are usually used in new DFS. Choosing a suitable file system and metadata storage system must be done during system design.

Other two methods, caching and file replication, can be added into the system later. Caching methods can increase system performance by predicting future requests. The most efficient caching policy seems to be LRU. File replication can increase system performance by spreading the system work-load to more servers. Caching and replicating algorithms can greatly increase system performance, but both of these algorithms do not always work reliably. On the one hand, these algorithms may increase system performance; on the other hand, if they are incorrectly set, system performance can be decreased.

VI. FUTURE WORK

In our future work we will participate in the KIVFS project. KIVFS is a distributed file system which is being developed at the Department of Computer Science and Engineering (Katedra Informatiky a Výpočetní techniky), University of West Bohemia. This distributed file system also considers usage of mobile devices. Accessing files from mobile devices requires algorithms which take into account changing communication channels caused by user's movement. We will develop caching and replication algorithms for this distributed file system.

VII. ACKNOWLEDGEMENTS

This work is supported by the Ministry of Education, Youth, and Sport of the Czech Republic – University spec. research – 1311. We thank Ladislav Pesička and Luboš Matějka, Doctoral students, Department of Computer Science and Engineering, University of West Bohemia, for their support and ideas.

REFERENCES

[1] Andrew S. Tanenbaum and Maarten Van Steen, *Distributed systems: principles and paradigms*. Upper Saddle River: Prentice Hall, 2002.

[2] (2002) Transparency in Distributed Systems. [Online]. http://crystal.uta.edu/~kumar/cse6306/papers/mantena.pdf

[3] Luboš Matějka, "Distributed File Systems," in *Computer Architecture and Diagnostic: Workshop for Doctoral Students: Lázně Sedmihorky*, 2005, pp. 125-128.

[4] T.Y.C. Woo and S.S Lam, "Authentication for distributed systems," *Computer* , vol. 25, no. 1, p. 39, January 1992.

[5] Rainer Többicke, "Distributed File Systems: Focus on Andrew File System/Distributed File Service (AFS/DFS)," in *Mass Storage Systems, 1994. 'Towards Distributed Storage and Data Management Systems.' First International Symposium. Proceedings, Thirteenth IEEE Symposium on*, Annecy, France, 1994, pp. 23-26.

[6] B. Liskov, R. Gruber, P. Johnson, and L. Shira, "A replicated Unix file system," in *Management of Replicated Data, 1990. Proceedings., Workshop on the*, Houston, 1990, p. 11.

[7] Lihua Yu, Gang Chen, Wei Wang, and Jinxiang Dong, "MSFSS: A Storage System for Mass Small Files," in *Computer Supported Cooperative Work in Design, 2007. CSCWD 2007. 11th International Conference on*, Melbourne, Australia, 2007, pp. 1087-1092.

[8] Lei Wang and Chen Yang, "TLDFS: A Distributed File System based on the Layered Structure," in *NPC '07 Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing Workshops*, Dalian, China, 2007, pp. 727-732.

[9] Pete Wyckoff Ananth Devulapalli, "File Creation Strategies in a Distributed Metadata File System," in *2007 IEEE International Parallel and Distributed Processing Symposium, 2007*, Long Beach, CA, USA, 2007, p. 105.

[10] P. Carns et al., "Small-file access in parallel file systems," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, Rome, Italy, 2009, pp. 1-11.

[11] D. Peric, T. Bocek, F.V. Hecht, D. Hausheer, and B. Stiller, "The Design and Evaluation of a Distributed Reliable File System," in *Parallel and Distributed Computing, Applications and Technologies, 2009 International Conference on*, Higashi Hiroshima, 2009, pp. 348-353.

[12] Bin Cai, Changsheng Xie, and Guangxi Zhu, "EDRFS: An Effective Distributed Replication File System for Small-File and Data-Intensive Application," in *Communication Systems Software and Middleware, 2007. COMSWARE 2007. 2nd International Conference on*, Bangalore, 2007, pp. 1-7.

[13] K. Shvachko, Hairong Kuang, S. Radia, and R. Chansler, "The Hadoop Distributed File System,", Incline Village, NV, 2010, pp. 1-10.

[14] J. Stender, B. Kolbeck, M. Hogqvist, and F. Hupfeld, "BabuDB: Fast and Efficient File System Metadata Storage," in *Storage Network Architecture and Parallel I/Os (SNAPI), 2010 International Workshop on*, Incline Village, NV, 2010, pp. 51-58.

[15] Benjamin Reed and Darrell D. E. Long, "Analysis of Caching Algorithms for distributed file systems," in *ACM SIGOPS Operating Systems Review, Volume 30 Issue 3*, New York, NY, USA, 1996, pp. 12-17.

[16] B. Whitehead, Chung-Horng Lung, A. Tapela, and G Sivarajah, "Experiments of Large File Caching and Comparisons of Caching Algorithms," in *Network Computing and Applications, 2008. NCA '08. Seventh IEEE International Symposium on*, Cambridge, MA, 2008, pp. 244-248.

[17] K.W. Froese and R.B. Bunt, "The effect of client caching on file server workloads," in *System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on*, Wailea, HI , USA, 1996, pp. 150-159.

[18] A. Ermolinskiy and R. Tewari, "C2Cfs: A Collective Caching Architecture for Distributed File Access," in *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*, Seoul, 2009, pp. 642-647.

[19] Lamprini Konsta and Stergios V. Anastasiadis, "Hades: Locality-aware Proxy Caching for Distributed File Systems," 2009.

[20] M. van Steen and G. Pierre, "Replication for Performance: Case Studies," in *Lecture Notes in Computer Science, Volume 5959*., 2010, pp. 73-89.

[21] S. Abdalla, I. Ahmad, Ewe Hong Tat, Gim Aik Teh, and Yong Lee Kee, "Towards Achieving a Highly Available Distributed File System," in *Advanced Communication Technology, The 9th International Conference on*, Gangwon-Do, 2007, pp. 2056-2060.

[22] Xin Sun, Jun Zheng, Qiongxin Liu, and Yushu Liu, "Dynamic Data Replication Based on Access Cost in Distributed Systems," in *Computer Sciences and Convergence Information Technology, 2009. ICCIT '09. Fourth International Conference on*, Seoul, 2009, pp. 829-834.

[23] (2008) CloudStore. [Online]. http://kosmosfs.sourceforge.net/features.html

[24] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google file system," in *SOSP '03 Proceedings of the nineteenth ACM symposium on Operating systems principles*, New York, NY, USA, 2003, pp. 29-43.

[25] (2010) Introduction to Gluster Versions 3.0.x. [Online]. http://download.gluster.com/pub/gluster/documentation/IntroductiontoGluster.pdf

Particle Swarm Optimization for Grid Scheduling

Jarmila Škrinárová and Michal Krnáč
Department of informatics FNS
Matej Bel University
Banská Bystrica, Slovakia
jarmila.skrinarova@umb.sk krnac.miso@gmail.com

Abstract—**The paper presents computational model for Grid scheduling and optimization criteria. The model of particle swarm optimizer (PSO) algorithm is described. We created simulation tool - Java applet (on the base of GridSim toolkit) which allows one to create and define model of multicomputer grid system, generate set of example tasks and simulate computation of this tasks. Used data are from real grid with its characteristics. There are 20 experiments and comparison between three optimize algorithms described.**

Keywords- grid scheduling, particle swarm optimization, modelling, simulation, web application, Java applet

I. INTRODUCTION

Job scheduling in its different forms is computationally hard. It has been shown that the problem of finding optimal scheduling in heterogeneous systems is in general NP-hard [6, 8]. Usually, an application can generate several jobs, which in turn can be composed of subtasks; the Grid system is responsible for sending each subtask to a resource to be solved. Grid systems contain schedulers that automatically and efficiently find the most appropriate machines to execute an assembly of tasks. Task represents a computational unit which runs on a Grid node. It is a program and possibly associated data. We can assume that a task is considered as an indivisible schedulable unit. Tasks could be independent (or loosely coupled) or there could be dependencies.

Job is a computational activity made up of several tasks that could require different processing capabilities and could have different resource requirements (CPU, number of nodes, memory, software libraries, etc.) and constraints, usually expressed within the job description. In the simplest case, a job could have just one task.

Grid scheduler is created by software components that are responsible for computing a mapping of tasks to Grid resources under multiple criteria and Grid environment configurations. Second part of this paper introduces Computational model for Grid scheduling and optimization criteria. There is a model of particle swarm optimizer (PSO) algorithm described in part 3. Simulation tool - a Java applet created on the base of GridSim toolkit in part 4. We decided to use data from real grid with its actual characteristics. We created 20 experiments on the base of PSO algorithm. In these experiments we evaluated and compared PSO with Hill Climbing and Round Robin algorithms. Experiments and used data are described in section 5.

II. COMPUTATIONAL MODEL FOR GRID SCHEDULING AND OPTIMIZATION CRITERIA

In this part computation model for Grid scheduling is presented. The computational capacity of the node depends on its:

- number of CPUs,
- amount of memory,
- basic storage space,
- other specifications.

The node has its own processing speed, which can be expressed in number of Cycles Per Unit Time (CPUT) [3].

A job is considered as a single set of multiple atomic operations (tasks). The task will be allocated to execute on one single node without pre-emption. The task has input and output data and processing requirements in order to complete its task. The task has a processing length expressed in number of cycles.

A schedule is the mapping of the tasks to specific time intervals of the grid nodes.

A scheduling problem is specified by:

- a set of machines,
- a set of jobs,
- optimality criteria,
- environmental specifications,
- other constraints.

One of the most popular optimization criteria is the minimization of the makespan. Makespan is an indicator of the general productivity of the Grid system. Small values of makespan mean that the scheduler is providing good and efficient planning of tasks to resources. Makespan indicates the time when the last task finishes and flowtime is the sum of finalization times of all the tasks.

Let *Jj (j{1,2,...,n})* be independent user jobs on *Gi (i{1,2,...,m})* heterogeneous grid nodes with an objective of minimizing the completion time and utilizing the nodes effectively.

Number of CPUT expresses speed of every node. The length of each job is specified by number of cycles. Each job *Jj*

has its processing requirement that can be expressed in number of cycles. A node *Gi* has its calculating speed and it is expressed in cycles/second. Any job *Jj* has to be processed in the one of grid nodes *Gi* until completion. Since all nodes at each stage are identical and preemptions are not allowed, to define a schedule it suffices to specify the completion time for all tasks comprising each job [1, 3].

To formulate our objective, we define:

- completion time *Ci,j (i{1,2,...,m}, j {1,2,...,n})* that the grid node *Gi* finishes the job *Jj*, represents the time that the grid node *Gi* finishes all the jobs scheduled for itself,
- makespan can be expressed as the *Cmax=max{∑Ci}*,
- flowtime is $\sum\left(\sum Ci\right)$.

An optimal schedule will be the one that optimizes the flowtime and makespan.

In our contribution we will minimize *Cmax*. It guaranties that no job takes too long. For minimization we choose PSO algorithm.

III. THE MODEL OF PARTICLE SWARM OPTIMIZER

The PSO algorithm was first applied to the optimization problems with continuous variables [4]. Recently, it has been used to the optimization problems with discrete variables [5, 7]. The optimization problem with discrete variables is a combination optimization problem which obtains its best solution from all possible variable combinations. The scalar *S* includes all permissive discrete variables arranged in ascending sequence. Each element of the scalar *S* is given a sequence number to represent the value of the discrete variable correspondingly. It can be expressed as follows [9]:

$$S_d = \{X_1, X_2, ...X_j, ...X_p\}, \qquad 1 \le j \le p$$

A mapped function *h(j)* is selected to index the sequence numbers of the elements in set *S* and represents the value *Xj* of the discrete variables correspondingly.

$$h(i) = X_j$$

Thus, the sequence numbers of the elements will substitute for the discrete values in the scalar *S*. This method is used to search the optimum solution, and makes the variables to be searched in a continuous space.

The PSO algorithm includes a number of particles, which are initialized randomly in the search space. The position of the *ith* particle in the space can be described by a vector x_i,

$$x_i = \left(x^{1_i}, x^{2_i}, ..., x^{d_i}, ..., x^{D_i}\right), \qquad 1 \le d \le D, \qquad i = 1, ...n$$

where *D* is the dimension of the particle, and *n* is the sum of all particles. The scalar corresponds to the discrete variable set *{X1, X2, ... , Xj, ... , Xp}* by the mapped function *h(j)*. Therefore, the particle flies through the continuous space, but only stays at the integer space. In other words, all the components of the vector *xi* are integer numbers. The positions of the particles are updated based on each particle's personal best position as well as the best position found by the swarm in every step.

The objective function is evaluated for each particle and the fitness value is used to determine which position in the search space is the best of the others. The swarm is updated by the equations (1) and (2).

$$V_i^{(k+1)} = \omega V_i^{(k)} + c_1 r_1 \left(P_i^{(k)} - x_i^{(k)}\right) + c_2 r_2 \left(P_g^{(k)} - x_i^{(k)}\right) \quad (1)$$

$$x_i^{(k+1)} = INT\left(x_i^{(k)} + V_i^{(k+1)}\right) \quad (2)$$

where $1 \le i \le n$, the current position is $x_i^{(k)}$ and the velocity is $V_i^{(k)}$ of each particle at the *kth* iteration, and $P_i^{(k)}$ is the best position in this particle (called *pbest*) and $P_g^{(k)}$ is the best global position among all the particles in the swarm (called *gbest*), *r1* and *r2* are two uniform random sequences generated from (0, 1), and ω the inertia weight used to discount the previous velocity of the particle persevered [8-10].

IV. GRID SIMULATION TOOL

In order to evaluate the model and simulate complex processes which run on grid systems we created a Java applet which allows one to create and define model of multicomputer grid system, generate set of example tasks and simulate computation of this tasks. The application uses several different algorithms to schedule tasks and their results are compared and displayed after finishing the simulation and gathering all the simulation data. Description of this project is placed on the web and the view is in "Fig. 1" [16].

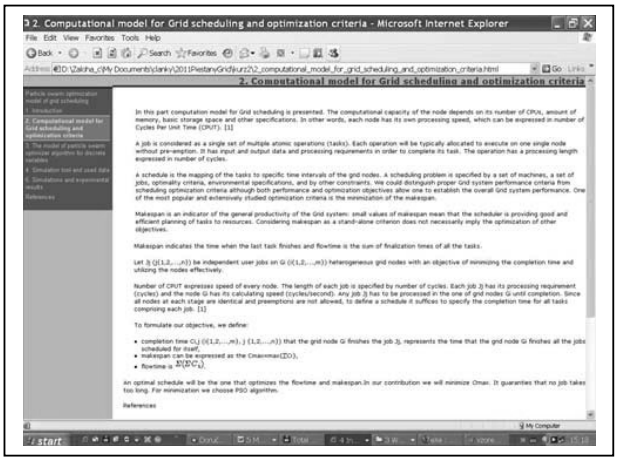


Figure 1. View of the research project

Applet uses external library of *GridSim* [11]. It provides engine for creating resources, describing network connections, creating users and processing the jobs they send to the system. Typical usage of the applet consists of several steps that are organized in tabs and follow each other in a logical order.

The *Grid settings* tab "Fig. 2" allows a user to create a grid resource and add it to a list of resources shown in left column. Bottom left buttons can be used to save or load current grid configuration to a user's hard drive. After clicking on the name of a resource, user can define basic resource characteristics: baud rate in Mbps, architecture, operating system and the cost of the system usage in dollars per second.

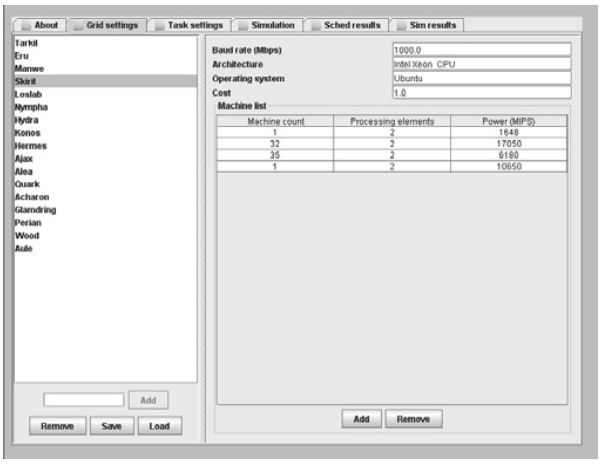


Figure 2. Setting parameters in applet

Each grid resource contains several machines which can be added to it in the right column. One row in the table represents all the machines of one kind that belong to a particular grid resource. Thus all the machines with the same number of processing elements and same computing power (processor speed) would be in one row.

Next step necessary to run the simulation is to define a set of tasks that will be simulated. This can be done on the *Task settings* tab. One can have program generated this set simply by clicking on *Generate tasks* button "Fig. 3". This action will use values provided in the form above the button.

Tasks can be then adjusted in the text area above. One row represents one task and its format is as follows: Length of the task in millions of instructions, input file size and output file size in bytes. All these values are integer numbers and separated by a space or tabulator. With all these values set we can start the simulation. It is done so on the *Simulation* tab. Because the primary aim of the application was to test PSO (Particle Swarm Optimization) scheduling algorithm, it provides options to set several parameters for this scheduler: swarm size, the number of iterations and to adjust weights as double numbers. After clicking on *Start simulation* tab the simulation starts and its process is shown in the text area below. However, the previous versions of *GridSim* did not allow running multiple simulations in one program. Therefore a user had to refresh the applet in order to repeat simulation more than one time.

At this point we recommend copying generated tasks to save it for using the same values again. After refreshing the applet, new tasks will be generated thus simulation may have different results. Last two tabs show results of the simulation "Fig. 4" in the form of a column diagram comparing PSO algorithm with Hill Climbing and Round Robin algorithms. Tab *Sched results* shows the times necessary to generate the schedule. Tab *Sim results* shows comparison of simulation times after using these schedules.

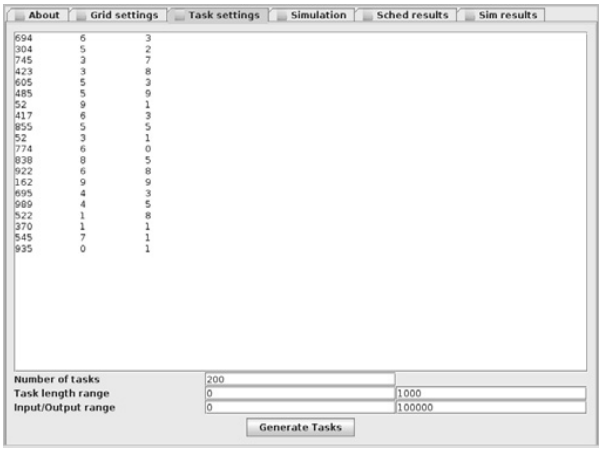


Figure 3. Task settings

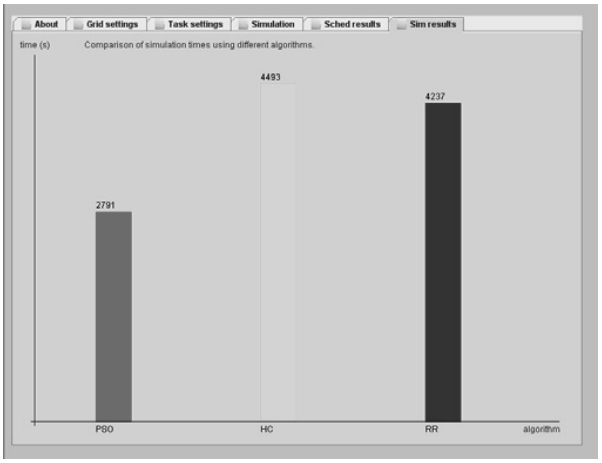


Figure 4. Comparison of simulation times after using schedules

V. EXPERIMENTS AND USED DATA

A. Used data

Data for grid simulation is taken from a grid monitor from the website of Nordugrid. The reason why we decided to use data from Nordugrid is, that our object is not to simulate randomly generated network of computers, but real functioning grid with all its characteristics.

The NorduGrid collaborative activity is based on the success of the project known as the "Nordic Testbed for Wide Area Computing and Data Handling", and aims at continuation and development of its achievements. That project was launched in May 2001, aiming to build a Grid infrastructure suitable for production-level research tasks [2].

The grid monitor in NorduGrid project operates and manages distributed computing infrastructure consisting of computing and storage resources owned by 10 countries as well as those of co-operative academic centers within the Europe. It is actively involved in many international Grid projects [2].

We choose some data from *SweGrid* "Tab. I" [13]. It is necessary to have information about resources and tasks. The resources data is composed of name of cluster, number of machines in cluster, number of processors, computer architecture and used operating systems. The task data can be expressed in number of tasks and number of input and output files. To be able to compare the efficiency of all computers, we chose to use a simple measure unit, which presents the number of floating point operations per second. Evaluation of every processor used is based on data taken from website of Geekbench benchmark [12].

TABLE I. EXAMPLE OF USED DATA

Name of resource	Resource information		
	Number of CPU	*Grid load*	*Number of tasks*
Beda	2032	752	117
Ritsem	528	246	1477
Ruth	316	308	1974
Siri	504	328	1150
Smokering	528	442	1266
Svea	496	431	1110

B. Experiments

We executed total of 20 experiments with different number of tasks "Fig. 5". The numbers of tasks were taken from Nordugrid monitor in the actual state. Every simulation input has tasks with large diversity of their length, which means that all task lengths are not evenly distributed in an input interval. Lengths of all tasks were generated randomly providing only minimum and maximum values of length, and file size. We used 10 particles and 300 iterations for PSO algorithm in every experiment.

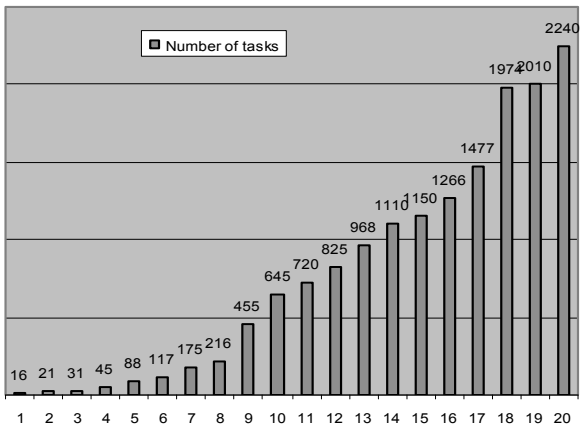


Figure 5. Numbers of tasks used in experiment

We applied PSO and Hill Climbing algorithm – which is believed to be the best scheduling algorithm so far – to every batch of tasks and observed total simulation time after using schedule generated by this algorithms. We also compared both of these algorithms with widely used round robin and its results. Additionally we recorded time which was needed to generate one schedule.

Schedule creation is an operation that affects total computing time, mainly when we use dynamic scheduling. However it is also important factor for static scheduling. The shortest time necessary for schedule creation has Round Robin and it is almost zero.

Result of experiments is shown in "Tab. II". Used PSO algorithms have the best total computing time even for large number of tasks. When we use small number of tasks the results are as follow: PSO schedule tasks slightly better than HC and far better than RR "Fig. 5". With large inputs PSO has almost the same efficiency as HC and very close to Round Robin. It confirms our previous experiments [15].

Known disadvantage of HC algorithm is, that when using large inputs it can result in one of possible local minimum instead of desired global minimum. The algorithm itself in its basic form has no option to decide whether the result is correct or not [14, 17].

TABLE II. RESULTS OF EXPERIMENTS

Num. of tasks	Total computing time (ms)		
	PSO	*HC*	*RR*
16	2157	2352	3200
21	3149	3929	4341
31	5269	5633	5217
45	6975	8871	10770
88	15269	17872	17175
117	20668	21007	26168
175	32138	34431	36981
216	41087	41004	46974
455	87133	97006	96628
645	120525	127792	141665
720	143200	148131	161249
825	165350	165348	182291
968	202555	205669	209238
1110	208839	211005	238436
1150	228565	229566	258547
1266	247229	255750	273854
1477	296247	305283	309801
1974	364498	364498	432682

Num. of tasks	Total computing time (ms)		
	PSO	*HC*	*RR*
2010	392796	392796	436459
2240	482479	482479	487981

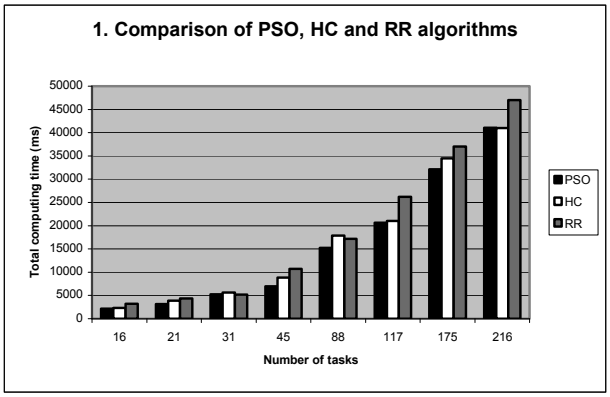


Figure 6. First part of comparison of computing time using PSO and other algorithms

.

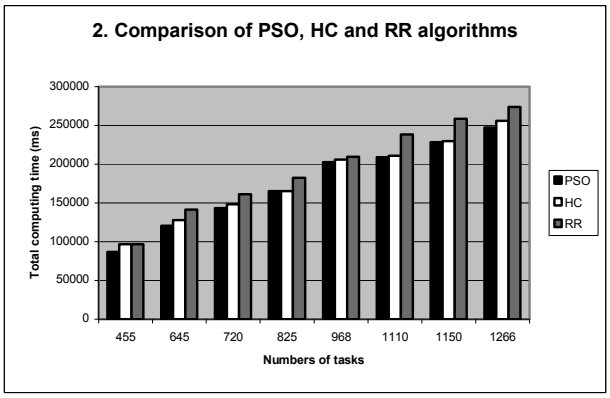


Figure 7. .Second part of comparison of computing time using PSO and other algorithms

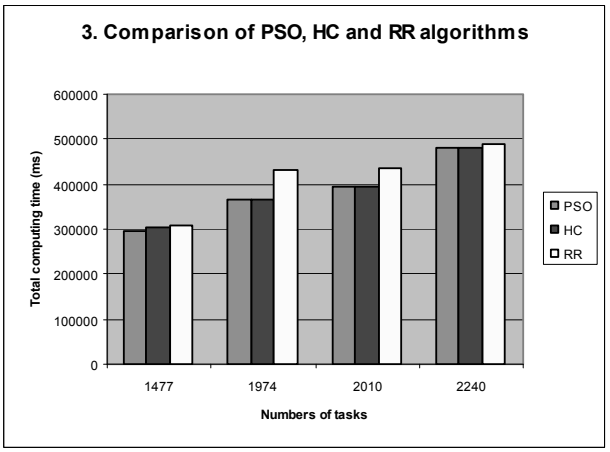


Figure 8. .Third part of comparison of computing time using PSO and other algorithms

VI. CONCLUSION

Implementing PSO algorithm brought improvement in grid scheduling optimization that shows low values of makespan. Results show that it is at least as good as popular and widely-used Hill Climbing algorithm which was considered to be the best until now. PSO algorithm is significantly better than the simplest Round Robin algorithm. These observations can be achieved thanks to the model of grid system and series of tests using this model. Model creation and its following evaluation by simulations is suitable technology for comparison with known scheduling algorithms

REFERENCES

[1] F. Xhafa, A. Abraham, "Computational models and heuristic methods for Grid scheduling problems." In: Future Generation Computer Systems. 26 pp. 608-621.

[2] F. Ould-Saada, "The NorduGrid Collaboration", http://www.nordugrid.org/about.html [Online; 1-September-2011].

[3] H. Liu, A. Abraham, A. Hassainen, "Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm", Future Generation Computer Systems.

[4] J. Kennedy, R.C., Eberhart, Y., Shi, 2001 "Swarm intelligence", Morgan Kaufman, San Francisco.

[5] L. J. Li, Z.B. Huang, F. Liu, 2007 "A heuristic particle swarm optimizer (HPSO) for optimization of pin connected structures", Computing Structure 85 (7–8) pp. 340–349.

[6] J. Škrinárová, M. Melicharčík, "Measuring concurrency of parallel algorithms", In: Proceeding of the 2008 1st International Conference on Information Technology Gdansk : IEEE computer society, 2008, pp. 289-292, IEEE Katalog Number: CFP0825E-PRT. ISBN 978-1-4244-2244-9.

[7] K.E., Parsopoulos, M.N., Vrahatis, 2002 "Recent approaches to global optimization problems through particle swarm optimization", Natural Computing 12, pp. 235–306.

[8] Y., Shi, R.C., Eberhart, 1997 "A modified particle swarm optimizer." In: Proceeding of IEEE congress on evolutionary computation;. p. 303–8.

[9] L.J., Li, Z.Y., Huang., F., Liu, 2009 "A heuristic particle swarm optimization method for truss structures with discrete variables." Computers & Structures, Volume 87 Pages 435-443

[10] S. He, Q.H., Wu, J.Y., Wen, 2004 "A particle swarm optimizer with passive congregation. Biosystem"; 78:135–47.

[11] R. Buyya, M, Murshed, "GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing", Concurrency and Computation: Practice and Experience, 2002, pp. 1175-1220.

[12] Geekbench, Geekbench result browser 2011 http://browse.geekbench.ca/, 2010.[Online; accessed 19-March-2011].

[13] F. Ould-Saada, "Grid monitor", http://www.nordugrid.org/. [Online; 1-September-2011].

[14] LUGER, G., F. 2001 Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 2001

[15] J. Škrinárová, M. Krnáč, "Particle Swarm Optimization Model of Grid Computing". In: 2th International Conference on Computer Modelling and Simulation, CSSIM 2011. Brno, Czech Republic: 2011, p. 146-153, ISBN: 978-80-214-4320-4

[16] J. Škrinárová, M. Krnáč, "E-learning course of scheduling of computer grid". In: 14th International Conference on Interactive Collaborative Learning, ICL 2011. Piešťany: 2011, p. 352-356, ISBN: 978-1-4577-1746-8©IEEE

[17] M. Hudec, "Genetic algorithms with adolescence and aging of genetic individuals." International conferenceI & IT 2009. Banská Bystrica FPV UMB, 2009. (In slovak)

Data Structures and Objects Relations in Virtual-reality System

Branislav Sobota, František Hrozek and Štefan Korečko
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics, Technical university of Košice,
Košice, Slovak Republic
branislav.sobota@tuke.sk, frantisek.hrozek@tuke.sk, stefan.korecko@tuke.sk

Abstract — **This paper describes areas of virtual-reality systems, specification of their structures, bindings and functions. The first part contains an introduction to different VR-systems and their division with specification of structures and bindings for particular frames. Possible parallel computations on parallel computer systems are described in the second part of the paper.**

Keywords - virtual reality, virtual objects, formal specifications, parallel computing.

I. INTRODUCTION

Virtual-reality (VR) system represents an interactive computer system that is able to create an illusion of physical presence in places in an imaginary world or in the real one. VR system can be also seen as providing a perfect simulation within the environment of tightly coupled human-computer interaction [1]. VR systems, day by day, provide more immersive experiences, they are more interactive, but the complexity of their implementation is raising, too. VR-subsystems categorization is accomplished especially according to senses which are affected by individual VR-systems parts [1]: *Visualization subsystem*, *Acoustic subsystem*, *Kinematic and statokinetic subsystem*, *Subsystems of touch and contact* and *Other senses* (e.g. sense of smell, taste, sensibility to pheromones, sensibility when being ill, pain, sleep or thoughts). Many of them are of little importance in virtual reality and they are explored in real life only to small extend, and thus, there is no use thinking about the simulation. Some of them have a meaning in real world, but we can question the possibility of their implementation in virtual reality. Taste belongs to such senses, but we do not expect virtual food to come into existence in future. From the point of view of VR-systems implementation, it is necessary to think about some of the above mentioned subsystems. Generally, data structures and objects relations in virtual-reality systems are very important for VR systems implementation. Description and scripting languages usage is needed in this case.

Description and scripting languages gained importance in areas of computer graphics and other IT systems [1][7][8]. Description languages such as *VRML* [9] are similar with HTML or XML languages. Scripting languages usually belong into group of interpreted languages. Main difference to compiled type of languages such as C++, C# or others is that script is executed from source code in most cases. Sometimes scripts are executed from pre-compiled code. As of today there are many kinds of description or scripting languages, for example *VRML*, *RUBY* [10], *PYTHON* [11] or *LUA* [12]. It is possible to define, program or control computer graphics or virtual-reality applications on higher level using appropriate language. Suitable data structures and objects relations and functions are the condition for next VR system implementation. Some of our ideas on appropriate data structures and objects relations and their usage in parallel VR system are presented in this paper.

II. SPECIFICATION OF VR ROLES FROM THE POINT OF VIEW OF THEIR SOLUTION

Specification of VR roles from the point of view of their solution in parallel system consists of following actions:

- input or processed information definition
- VR objects definition and placement of individual input information into these VR objects
- creation or completion of these VR objects by relations and database (relations and database describe features of whole virtual world and also of individual VR objects, and the objects in the interrelation to other objects)
- specification of VR objects processing methods, i.e. operations specification for individual VR objects - encapsulation (from the point of view of object programming)
- specification of possibility of parallel computation of individual VR objects.

The issue of interaction level between the observer and the environment is of great importance. VR system of DEDO (Dynamic Environment Dynamic Observer) type is the most interesting for us (because of the simplicity we consider just one observer) as we intend to use a fully immersive VR model.

A. Specification of input and processed information

Because we deal with fully interactive systems, it is necessary to acquire information about movement of a human or an observer in the virtual world. This information is being acquired from kinematic and statokinetic subsystems (e.g.

position trackers or data gloves). It is not necessary to consider the form in which individual information is acquired in this phase of specification; however, it is necessary to name them. They are as follows:

- information concerning the position of all the rotational joints on human body (e.g. position of head, hands, individual finger phalanxes position, position of body etc.)
- information about observer view direction

Further information is being acquired from global information of VR object (the term global information of VR object will be explained in the following section):

- position, information about location of individual objects in virtual world from the point of view of their interaction with other objects or observers
- material features, features of individual objects that are necessary to take into consideration when speaking about interaction between objects or between object and observer (it concerns mainly the so called feedback, i.e. what sort of sound should be generated for the given object after the interaction with another one, how it should be deformed, etc.)

The design of the virtual world representation is a non-trivial problem because it determines the implementation of the VR system as well. We need the best possible implementation providing fast modification, visualization and information retrieval [2][3].

A possible structure is shown in Figure 1. The object (group of objects, world) is described by static objects and sensor objects and characteristic frames. Other attributes are included too.

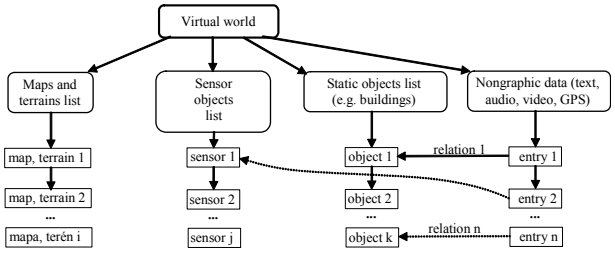


Figure 1. Virtual world elements

Three-dimensional area modeling used in VR system is not yet a well-researched field [4]. The first question arising is whether to model the whole environment as one complex entity into details or to create a less concrete model and push the detail processing to the phase of later actualization [5]. Next important question is about the way of model design. There are alternatives as follows:

- using existing concept material
- creation of a model from a new data base
- combination of the previous two alternatives

The problem of the first alternative is in the actuality of the data in the existing materials, and of course, these data are aged and therefore the quality might be less than expected too. So the errors might be included in the representation as well.

The second alternative solves these problems, but is much more expensive as being really executable. Laser scanning of large areas and satellite scans are not yet economic.

Because it is also necessary to integrate non-graphical data into the VR system, the combination of the methods is the most effective and realistic option.

Besides mentioned scan techniques, terrain can be modeled using polygons too. On such a surface textures are also applicable. Another good approach is the use of geodetic surface maps (vertical coordinates are represented by different colors), which allows automatic surface model generation; textures are applicable to that model as well.

Sensors are objects used with collision detection and they are useful in the case of interactive or automated presentations of the virtual reality (VR) scenarios. Events on sensors are of two kinds:

- Stop move if collision occurs
- Change move direction if collision occurs

The use cases of sensor objects are:

- Avoid camera moves into/across VR objects
- Display non-graphical information about the VR object when entering the checked area

Collision detection is executed on the scene objects (static and sensor objects) and the camera represented as a sphere object. Depending on the kind of the object in collision, the VR system stops the camera and/or displays additional information related to the selected static or sensor object.

In general, we do not have to assign a sensor to each static object, and the system also might have more types of sensors implemented. The types of the sensors are as follows:

- *Vertical sensor polygon* – detection of manual object selection in the case that appropriate tools (e.g. mouse) are used.
- *Horizontal sensor polygon* – camera position detection, status is active if collision between camera and other object occurs.
- *Point-of-view sensor* – status is active if the object is inside the view frustum and it is the closest one to the camera (observer).

B. VR object definition and placement of individual input information into VR objects

The whole VR system can be defined as strictly object system. This means that everything that exists in the VR system is an object (maps, buildings, sensors etc.). Even the VR system itself can be defined as object which contains other objects.

In terms of such definition it is possible to exploit the features of inheritance, encapsulation and polymorphism. In Figure 2 hierarchy of objects in terms of VR-system is shown.

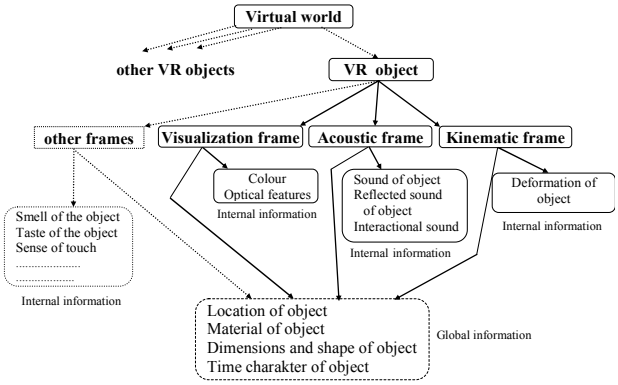


Figure 2. VR object frames from the definition point of view

As it can be seen in Figure 2, VR object contains frames which are divided according to basic subsystems of VR system. Each of these frames contains internal information invisible for other frames. For frames of VR object global information exist as well. It is necessary for each of these frames, so it is shared.

Virtual object implementation naturally implies parallel problem solution by means of computer with parallel architecture. However, the problem concerning creating data structures is very extensive. Its optimization is of great importance. It is obvious when we consider the following:

- the whole system consists of great number of objects
- each of these objects must be defined in terms of its:
  - shape and visual features
  - mechanical features
  - sound features
  - time characteristics

C. *Relations definition between objects*

It is of great importance to correctly define features of a virtual object since they closely specify the object itself. They define its behavior, its look and sounds; all being done in terms of situation "without any conflict" and in interaction with other object as well. Basically, two kinds of representation of these data are possible:

- representation by means of database: If all these features are created by database (in which it is for example defined in what way the given object behaves in interaction with another one having certain features), it is clear that these bases could be infinite.
- representation by means of relations: All features are defined by means of the so called relational equation which defines on the basis of certain regulation (term) how the object should behave in different situations or how it will be deformed when affected by external power etc.

As could be seen from the previous, the only appropriate way is to use a combination of database and relational equations. One of the possible realisations would be for example representation of internal features of an object by means of database (e.g. from which material the object consists, optical features etc.), and features in interaction with other objects would be represented by relations (characteristics of what shall happen in collision with VR object of certain type etc.). A scripting language is usable in this case.

D. *Definition of operations with VR objects*

It is necessary to define operations with individual objects, i.e. the way in which it is possible to treat individual VR objects. We can for example define binary operations as creating of more complex objects by means of simpler ones, concerning either shape, colours, sound or something else.

For example, if we name operator "~" composition of objects then operation

```
VRObject3 = VRObject1 ~ VRObject2
```

creates more complex VRObject3 which contains features of both the previous objects.

There exists a possibility to use operations or operators which are defined for each of the objects in the same way, however, the implementation varies. Both operations have the same syntactical record, however, their implementation part, i.e. processing itself, varies. Scripting language is usable on the implementation level for appropriate operator.

III. POSSIBILITIES OF VR OBJECTS PARALLEL COMPUTING

Levels of parallelism can be as follows [1]:

- parallelism on the level of VR realities - parallel processing of more virtual worlds
- parallelism on the level of subsystems - parallel processing of individual VR system subsystems such as visualization, kinematic and acoustic.
- parallelism on the level of objects - parallel processing of individual VR world objects.
- parallelism on the level of frames - parallel processing of individual object frames.
- parallelism on the level of algorithms - e.g. parallel computing in individual frames etc.

Thus, according to these levels we can graduate granularity of parallelism beginning from coarse-grained (worlds level) to fine-grained (algorithms level).

Parallelization on worlds level and subsystems level is trivial. The fact, that objects in VR system are relatively independent, allows us to consider their parallel processing. This means, that each object can be considered an independent branch. The following graph shows possible parallel solutions.

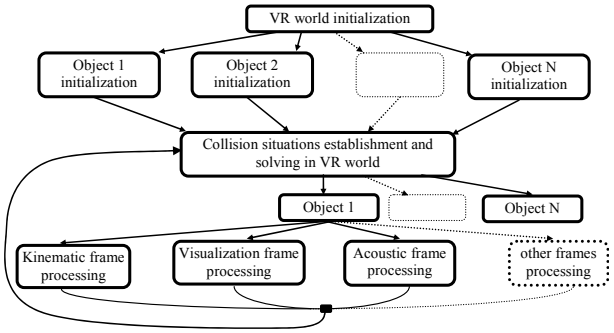


Figure 3. The parallel implementation of virtual objects and their frames

As it can be seen, this is highly parallel system. If we take Figure 3 for our starting point, it is possible to make the initialisation of VR world with "N" objects parallel, and processing of individual VR objects itself as well. It is also possible to create parallel computing branches in terms of these objects. For example: the action of the part "Collision situations establishment and solving in VR world" can be made in parallel as follows (Figure 4).

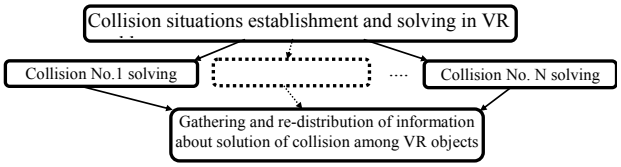


Figure 4. The parallel computing of "Collision situations establishment and solving in VR world" block

For parallelism on the frames level it is obvious, that possibilities of making parallelisms vary considering different character of individual frames, nevertheless, in general, it is possible to exploit parallel processing in each of them. As an example serves "kinematic frame" and its parallel computing consists of following steps:

- Object information input (e.g. position, time etc.)
- Parallel computing of new object position (parallel matrix computing)
- Writing of new parameters into internal or global variables of object

It is hard to deal with algorithms level parallelism in general as individual algorithms differ significantly.

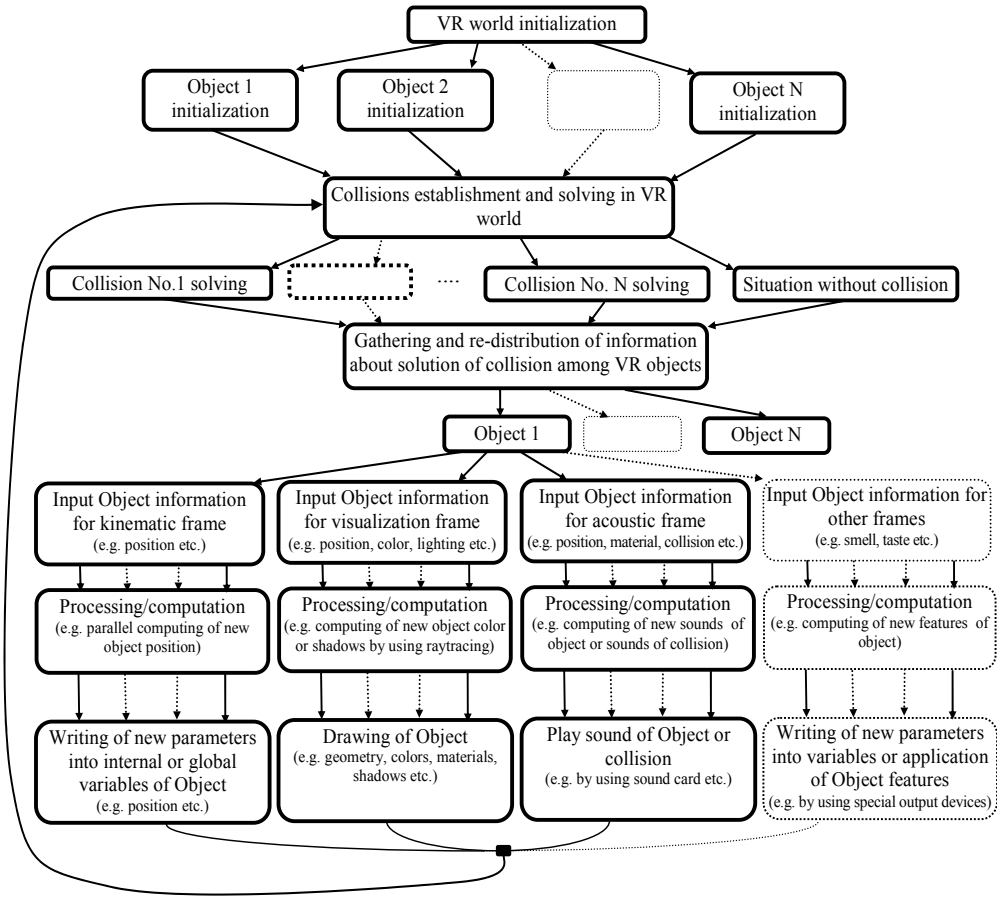


Figure 5. The global structure of VR-system parallel computing

IV. CONCLUSION

Described VR system structure and parallel processing has been partially implemented in the VR system developed at the DCI FEI TU Košice. This system provides mainly immersive stereoscopic visualization of complex datasets [1]. Typical dataset represents a part of the Košice city or a historical building from another location (e.g. Spis castle) [6]. Implemented system not only provides nicely textured geometry, but also some information about objects. This VR system can be used like fully immersive interactive information system working in real time (Figure 6). It has high hardware requirements and one solution is parallel implementation. It is obvious from the previous figure (Figure 5), that the problem concerning VR roles processing in VR systems is highly parallel issue. VR-systems that can process such almost independent branches in parallel are appropriate to solve this problem.

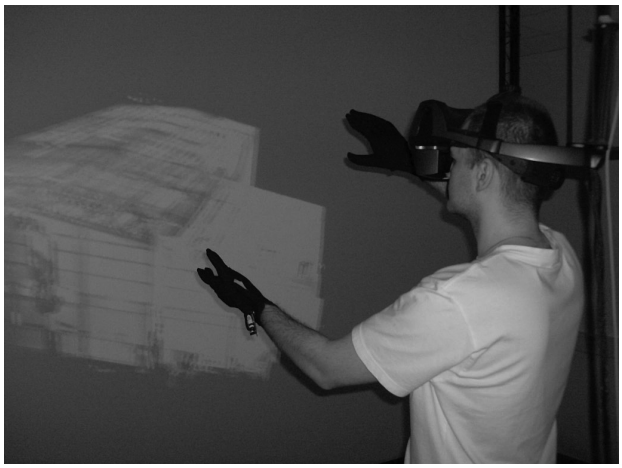


Figure 6. User working with VR system through interaction with virtual world (DCI FEEI TU Košice)

ACKNOWLEDGMENT

This work is supported by VEGA grant project No. 1/0646/09: Tasks solution for large graphical data processing in the environment of parallel, distributed and network computer systems.

REFERENCES

[1] Sobota, B., Perháč, J., Straka, M., Szabó, Cs.: Applications of parallel, distributed and network computer systems for solving of computational processes in an area of large graphical data volumes processing; elfa Košice, 2009, ps. 180, ISBN 978-80-8086-103-2 (in slovak)

[2] Sobota, B., Straka, M., Sobotová, D.: 3D rozhranie informačného systému. Informatika a informačné technológie I&IT '04, Banská Bystrica, Vydavateľstvo Bratia Sabovci, Banská Bystrica, 2004, pp. 52-56

[3] Vokorokos, L., Perháč, J., Kleinová, A.: Parallel Computer System Utilization in Data Visualization. Informatics' 2007, Bratislava, Slovakia, 2007

[4] Sobota, B., Straka, M., Hlinka, F., Perháč, J.: Parallel processing of visualization of 3D virtual map project, MOSMIC 2007 - Modelling and Simulation in Management, Informatics and Control, Žilina, 2007, pp. 9-14

[5] Sobota, B. – Perháč, J. – Petz, I.: Surface modeling in 3D city information system, Journal of Computer Science and Control Systems, 2, 2, 2009, pp. 53-56, ISSN 1844-6043

[6] Sobota Branislav, Korečko Štefan, Perháč Ján: 3D Modeling and Visualization of Historic Buildings as Cultural Heritage Preservation, Proceedings of the Tenth International Conference on Informatics INFORMATICS 2009, Herľany, Slovakia, November 23-25, 2009, Košice, Slovakia, elfa, 2009, 10, 1, pp. 94-98, ISBN 978-80-8086-126-1

[7] Yangl, X., Petriu, D.C., Whalen, T.E., Petriu, E.M.: Script Language for Avatar Animation in 3D Virtual Environments, VECIMS 2003 - lntemalional Symposium on Virtual Environments, Human Compuler Interfaces, and Measurement Systems, Lugana, Switzerland, 27-29 July 2003, pp. 101-106

[8] Dietrich A., Gobbetti E., Yoon S.-E., ″Massive-Model Rendering Techniques: A Tutorial″, IEEE Computer Graphics and Applications, vol. 27, no. 6, pp. 20-34, 2007.

[9] VRML – Virtual Reality Modeling Language - http://www.w3.org/MarkUp/VRML/

[10] Ruby official website, url: http://www.ruby-lang.org/en/

[11] Python Programming Language – official website, url: http://www.python.org/

[12] LUA - the programming language – official website, url: http://www.lua.org/

Preservation of Historical Buildings Using Virtual Reality Technologies

František Hrozek, Branislav Sobota and Csaba Szabó
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics, Technical university of Košice
Košice, Slovak Republic
frantisek.hrozek@tuke.sk, branislav.sobota@tuke.sk, csaba.szabo@tuke.sk

Abstract — **3D model creation and visualization is often used in preservation of historical buildings. Virtual reality (VR) technologies facilitate this 3D model creation and visualization. 3D scanner can be used for acceleration of the data collection and model creation. 3D displays can be used to improve visual perception and 3D printing can be used for physical model creation. In this paper is presented the way for preservation of historical buildings using virtual reality technologies. Example of preservation procedure, which consists from 3D scanning, 3D model creation, optimization and visualization, is shown on the historical building of The State Theatre of Košice.**

Keywords - virtual reality technologies, 3D scanning, 3D modeling, 3D visualization, preservation of historical buildings

I. INTRODUCTION

Preservation of historical buildings is important thing in cultural heritage preservation. In the past was for historical building preservation used blueprints, sketches or paintings. Today technologies allow us to create 3D models of historical buildings that can be used for preservation.

Historical building 3D model creation needs a lot of effort. Everything begins with collecting of information and analysis (preparing phase). When the data are prepared 3D model creation begins (modeling phase). A check of model for errors comes after 3D digital model creation (verification phase). The visualization of the final model is the last step. This process is depicted in Fig. 1 [1].

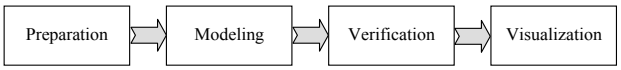


Figure 1. Modeling and visualization process

Modeling and visualization phase can be improved by using virtual reality technologies. Modeling phase can be improved with 3D scanning and visualization phase with 3D displays and 3D printers.

In this paper is presented the way for preservation of historical buildings using VR technologies. Section 2 presents 3D scanning problematic. In section 3 is mentioned scanning process of The State Theatre of Košice. Section 4 presents VR technologies that can be used for visualization. Section 5 summarizes information presented in the paper.

II. 3D MODEL AQUISITION PROCESS

The 3D model acquisition process (Fig. 2) consists of two main stages which are [2]:

- 3D scanning
- data processing

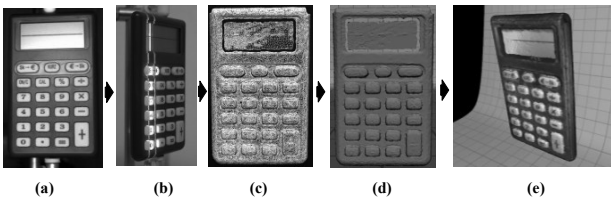


Figure 2. 3D model acquisition process: (a) original object, (b) scanning process, (c) scanned object (point cloud), (d) model without texture, (e) visualized final textured object.

A. 3D Scanning

In general, the result of a 3D scan is a set of points in space with corresponding 3D coordinates called *point cloud* (Fig. 2c). To capture the whole object, a series of scans from various angles has to be made. There are several types of 3D scanners, which differ in the technology used for obtaining a point cloud. They can be divided into two main categories:

- contact scanners
- non-contact scanners

1) Contact scanners

Contact scanners require a physical contact with the object being scanned. Although they are usually very precise, they are also much slower (order of 10^3 Hz) than non-contact scanners (order of 10^5 Hz). A typical example of a contact 3D scanner is a coordinate measuring machine (CMM).

2) Non-contact scanners

Non-contact scanners use radiation to acquire required information about objects. They are of two basic types: *passive* and *active*. The main advantage of passive scanners is that they are cheap as they do not require so specialized hardware to operate. To scan objects, they only use existing radiation in its surroundings (usually visible light). In contrast, active

scanners are equipped with their own radiation emitter (usually emitting laser light). While the latter are considerably more expensive, they are also much more accurate and able to scan over much bigger distances (up to few km).

B. Data Processing

Data processing consists of several parts. First, the point cloud has to be *meshed*, i.e. the points have to be connected into a collection of triangles (called *faces*). The next step is to *align* the scans from various angles to create the whole object surface. The aligned scans then have to be *merged* into one continuous *mesh*, so that no overlapping parts occur. The merging process also involves filling the eventual "holes" (unscanned parts) in the model. Aligning and mesh creation step can be switched if it is required for data processing. Additionally, there is an optional step to *simplify* the mesh, which consists of reducing the number of triangles in order to save memory needed to visualize the final 3D model.

C. Used Scanner

For scanning was used 3D scanner Leica Scanstation 2 (Fig. 3) from Leica Geosystems. More details about this 3D scanner (e.g. scanning range, scanning rate or scanning angles) at manufacturer's webpage [3].

Figure 3. Leica ScanStation 2

III. SCANNING OF THE STATE THEATRE OF KOŠICE

A. 3D Scanning

For scanning of any object it is important to find right scanning positions to acquire the best 3D scans with the minimum number of scans. Optimal number of scans for buildings that has rectangular ground-plan (like the State Theatre of Košice) is eight. Four scanning positions are perpendicular to the walls (Fig. 4, positions: 1, 3, 5, 7) and another four scanning positions lies approximately on the diagonals of building (Fig. 4, positions: 2, 4, 6, 8). Schematic view of scanning positions for the State Theatre of Košice are shown on Fig. 4. Height of scanner for all eight scanning positions was 1.8 m and distance to theatre was approximately 20 m.

Scanning process consisted from these steps (all steps except first one were made in Cyclone):

- placing markers (Fig. 5) (these markers serves for scans aligning; to align two scans at least 3 markers are needed that are mutual for both scans)
- setting horizontal and vertical angle for scanning
- create a photography of scanned object (this photography serves as texture for scanned object)
- setting scanning resolution (for scanning was used resolution 2 cm × 2 cm; this resolution was selected after resolution tests and offers good scan details with reasonable scanning time)
- 3D scanning
- acquiring exact positions of markers

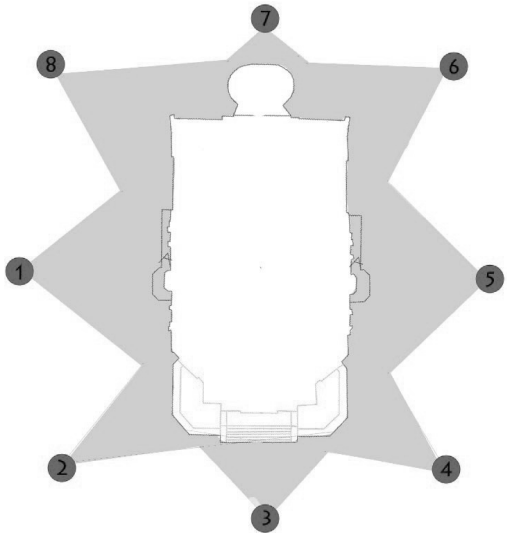


Figure 4. Schematic view of scanning possitions

Figure 5. Markers used for scans alighning

Photography creation time, scanning angle and time of every scanning position are shown in the following table (Tab. 1). Textured point cloud scanned from position 5 is shown on Fig. 6.

TABLE I. SCANNING TIMES AND ANGLES

Scanning Position	Photo Creation Time	Horizontal Angle	Vertical Angle	Scanning Time
1	2m 5s	100°	43°	9m 35s
2	2m 1s	62°	45°	5m 29s
3	1m 56s	78°	51°	8m 5s
4	1m 30s	50°	45°	4m 15s
5	2m 2s	105°	43°	10m 35s
6	1m 20s	42°	31°	3m 44s
7	1m 25s	68°	34°	5m 59s
8	1m 24s	35°	27°	4m 30s

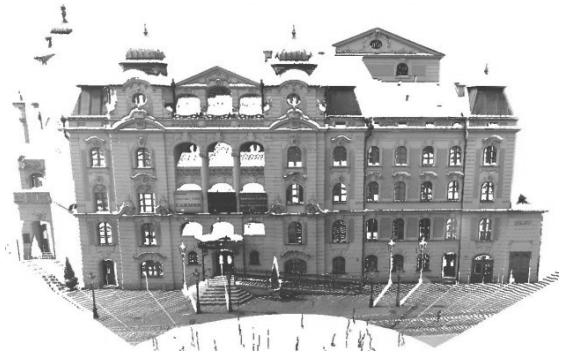


Figure 6. Point cloud with texture (position 5)

B. Data processing

Data processing was divided into parts:

- aligning point clouds and mesh creation
- mesh simplifying

1) Aligning point clouds and mesh creation

For point clouds aligning was used Cyclone (Fig. 7 shows part of Cyclone that serves for aligning). Point clouds can be aligned manually or automatically. For manual aligning it is necessary to pick at least 3 points that are mutual for both aligned point clouds. Automatic aligning is similar to manual (also at least 3 point are necessary) but points are taken automatically from coordinates taken by scanning markers. For point clouds aligning of The State Theatre of Košice was used automatic aligning.

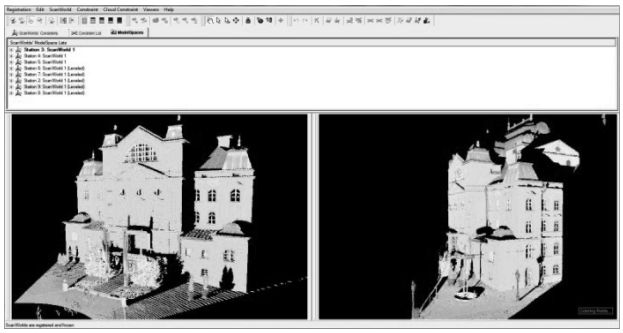


Figure 7. Aligning point clouds using Cyclone from position 7 and 8

Fig. 8 shows final point cloud created from all eight point clouds. Final step of this part was mesh creation from point clouds using Cyclone.

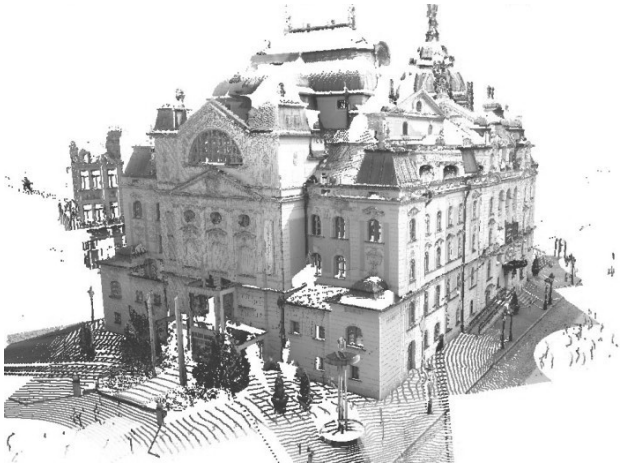


Figure 8. Final point cloud of The State Theatre of Košice after merging

2) Mesh simplifying

For mesh simplifying was used 3D modeling application 3ds Max [4]. Simplifying consisted from two parts:

- deleting meshes that did not belong to theatre (for example parts of surrounding buildings that was scanned with theatre – Fig. 9)

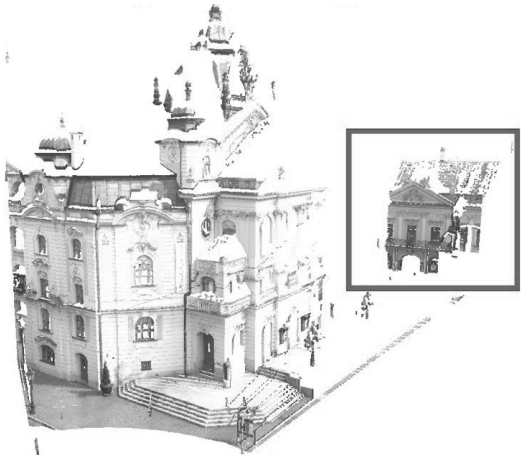


Figure 9. Example of mesh that was deleted

- simplifying theatre mesh using 3ds Max *Optimize* function. For *Optimize* function was used default setting, only parameter *Face thresh* was set to 6.0. This setting was selected after tests and offers good ratio between mesh details and vertices/faces number.

By deleting meshes that did not belong to the theatre and using *Optimize* function was achieved reduction to 3 715 977 vertices (63.3%) and 5 814 551 faces (56.5%). Tab. 2 shows vertices and faces count before and after simplifying.

TABLE II. VERTICES AND FACES BEFORE/AFTER SIMPLIFYING

Scanning Position	Before Simplifying		After Simplifying	
	Vertices	*Faces*	*Vertices*	*Faces*
1	472 333	807 419	244 095	341 884
2	923 648	1 630 899	609 886	1 005 252
3	1 500 267	2 661 869	1 240 775	2 144 765
4	768 476	1 373 441	515 857	880 426
5	819 735	1 413 898	445 570	664 532
6	270 690	460 492	128 263	179 461
7	429 621	730 319	204 790	271 490
8	680 920	1 208 556	326 741	326 741
Entire model	5 865 690	10 286 893	3 715 977	5 814 551

IV. VISUALIZATION WITH VR TERCHNOLOGIES

A. 3D Displays

3D displays allow information presentation in three dimensions. Exists several technologies for 3D displaying and each technology has its advantages and disadvantages. There are several types of 3D displays that can be used for 3D visualization [5].

DCI FEEI TU of Košice use for 3D visualization passive stereoscopic system and autostereoscopic 3D display.

1) Passive stereoscopic system

This system uses passive stereoscopic technology based on INFITEC technology (more details about INFITEC technology at [6]). Parts of this system are (Fig. 10):

- pair of projectors with INFITEC filters and glasses
- special projection screen
- mouse or "space mouse" for navigation in 3D scene
- rendering cluster consisting from three PCs with cluster version of visualization software called SuperEngine

Rendering cluster currently supports scenes rendering up to 5 million polygons. Rendering performance of this cluster can be extended by adding other computers to cluster.

Figure 10. Parts of stereoscopic system: screen, glasses and projectors

2) Autostereoscopic 3D display

This autostereoscopic 3D display is Philips WOWvx (Fig. 11). 3D image is created using 2D-plus-depth method. More details about this 3D display and 2D-plus-depth method at manufacturer's webpage [7]. Fig. 11 shows 3D visualization of The State Theatre of Košice using Philips WOWvx.

Figure 11. 3D visualization using Philips WOWvx

B. 3D Printers

3D printing is a form of additive manufacturing technology where a three dimensional object is created by laying down successive layers of material. 3D printers are generally faster, more affordable and easier to use than other manufacturing technologies. 3D printers offer developers the ability to print 3D models for visualization, testing or direct parts creation.

3D printer at DCI FEEI TU of Košice is ZPrinter 450 (Fig. 12 left). More details about this 3D printer (e.g. maximum printing dimensions or printing resolution) at manufacturer's webpage [8]. Fig. 12 (right) shows 3D model of St. Michael Chapel which was printed with ZPrinter 450. Work on the printed model of The State Theatre of Košice is currently underway.

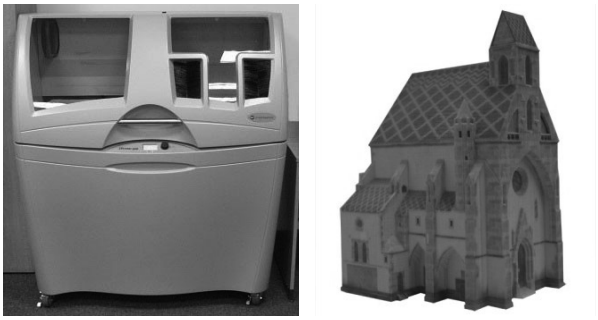


Figure 12. ZPrinter 450 (left) and printed model of St. Michael Chapel (right)

V. CONCLUSION

The aim of this paper was presentation of historical building modeling and visualization for cultural heritage preservation using virtual reality technologies. Example of preservation procedure, which uses 3D scanning, 3D model creation, optimization and visualization, is shown on the historical building of The State Theatre of Košice. For

scanning was used 3D scanner Leica ScanStation 2. Point clouds were aligned and meshed using Cyclone software. Final mesh (3D model) was simplified using 3ds Max. Final model have 3 715 977 vertices and 5 814 551 faces. For visualization purposes was presented these VR technologies:

- 3D displays – passive stereoscopic system with INFITEC technology and autostereoscopic 3D display Philips WOWvx
- 3D printing – ZPrinter 450

Presented information shows that VR technologies are very suitable for creation and visualization of 3D models used in preservation of historical buildings. Also VR technologies simplify 3D model creation and visualization process.

Future work will be focused on:

- implementation of other VR technologies into historical building preservation (e.g. augmented reality or head mounted displays)
- preservation of other historical buildings in Slovak Republic

ACKNOWLEDGMENT

This work is supported by VEGA grant project No. 1/0646/09: Tasks solution for large graphical data processing in the environment of parallel, distributed and network computer systems.

REFERENCES

[1] Sobota Branislav, Korečko Štefan, Perháč Ján: 3D Modeling and Visualization of Historic Buildings as Cultural Heritage Preservation, Proceedings of the Tenth International Conference on Informatics INFORMATICS 2009, Herľany, Slovakia, November 23-25, 2009, Košice, Slovakia, elfa, 2009, 10, 1, pp. 94-98, ISBN 978-80-8086-126-1

[2] Sobota Branislav, Rovňák Maroš, Szabó Csaba: 3D scanner data processing, Journal of Information, Control and Management Systems, Volume 7, 2, 2009, pp. 191-198, ISSN 1336-1716

[3] Leica ScanStation 2 homepage, [online], [cited 2011-28-8]. Available on internet: <http://hds.leica-geosystems.com/en/Leica-ScanStation-2_62189.htm>

[4] Autodesk 3ds Max homepage, [online], [cited 2011-30-8]. Available on internet: <http://usa.autodesk.com/3ds-max/>

[5] Xu, S.; Manders, C.M.; Odelia, T.Y.; Song, P.: 3D display for a classroom, Educational and Information Technology (ICEIT), 2010 International Conference on, vol. 2, pp. V2-316-V2-320, 17-19 September 2010

[6] INFITEC homepage, [online], [cited 2011-29-8]. Available on internet: <http://www.infitec.de/index2.php>

[7] Philipse WOWvx homepage, [online], [cited 2011-29-8]. Available on internet: <http://www.business-sites.philips.com/3dsolutions/home/index.page>

[8] ZCorp ZPrinter 450 homepage, [online], [cited 2011-30-8]. Available on internet: <http://zcorp.com/en/Products/3D-Printers/ZPrinter-450/spage.aspx>

An Overview of Ontology Learning From Unstructured Texts

Marco Alfonse Tawfik, Mostafa M. Aref, Abdel-Badeeh M. Salem
Computer Science Department, Faculty of Computers & Information Sciences, Ain Shams University
Cairo, Egypt
E-Mail: marco_alfonse@cis.asu.edu.eg, aref_99@yahoo.com, absalem@asunet.shams.edu.eg

Abstract - **An ontology is an explicit, formal specification of a shared conceptualization of a domain of interest, where formal implies that the ontology should be machine-readable and shared means it is accepted by a group or community. Further, it should be restricted to a given domain of interest and therefore model concepts and relations that are relevant to a particular task or application domain. Thus a fast and efficient ontology development is a necessity. However, ontology development is a difficult and time consuming task. Ontology learning can be defined as the set of methods and techniques used for building an ontology from scratch, enriching, or adapting an existing ontology in a semi-automatic fashion using several sources. These sources can be databases, structured and unstructured documents or even existing preliminaries like dictionaries, taxonomies and directories. Here, the focus is on the acquisition of ontologies from unstructured text, a format that scores highest on availability but lowest on accessibility. This paper provides a discussion on existing ontology learning techniques from unstructured text and the state of the art of the field.**

Keywords-ontology; semantic web; knowledge representation; ontology learning.

I. INTRODUCTION

The term ”ontology” is concerned with a theory about the existence. The Artificial Intelligence discipline considers ontologies as a formal specification of the concepts of an interest domain, where their relationships, constraints and axioms are expressed, thus defining a common vocabulary for sharing knowledge. An ontology can be defined as a formal, explicit specification of a shared conceptualization [1].

One of the greatest applications of ontologies is the Semantic Web [2], a new generation of the Web in which the semantic of documents would be expressed using ontologies. This way, the Semantic Web is an approach for enhancing the effectiveness of Web information access. However, the manual construction of ontologies is an expensive and time consuming task because the professionals required for this task (i.e. a domain specialist and a knowledge engineer) usually are highly specialized. This difficulty in capturing the knowledge required by knowledge based systems is called “knowledge acquisition bottleneck”. The fast and cheap ontology development is crucial for the success of knowledge based applications and the Semantic Web. An approach for this problem is to provide an automatic or semi-automatic support for ontology construction. This field of research is usually referred to as ontology learning. This paper presents a survey of ontology learning techniques from unstructured texts. Section 2 introduces the literature studies. Section 3 presents the methods used for ontology learning from unstructured texts. Section 4 provides the methods used for evaluating ontology learning approaches. Section 5 presents some ontology learning tools from text and finally section 6 contains the conclusions.

II. LITERATURE STUDIES

There is no standard regarding ontology development process, the aspects and tasks involved in ontology development can be viewed into a set of layers [3]. An ontology consists of concepts, relationships between them and axioms. In order to identify the concepts of a domain, in first place, it is necessary to identify the natural language terms that refer to them. Synonym identification helps to avoid redundant concepts, since two or more natural language terms can represent the same concept. Another reason for identifying the terms is that, in the future, some of them can be used to uniquely identify their respective concepts. The terms are the source for identifying the concepts that will be part of the ontology. The next step is to identify the taxonomic relationships (generalization and specialization) between the concepts (concept hierarchies). It is also necessary to extract the non-taxonomic relations, thus defining the set of relationships. Finally the extraction of the instances of the learned concepts and relationships takes place. Some authors also consider rule acquisition for deriving facts that are not explicitly expressed in the ontology.

The term ontology learning [4] refers to the automatic or semi-automatic support for the construction of an ontology, while the automatic or semi-automatic support for the instantiation of a given ontology is referred to as ontology population. Ontology learning is concerned with knowledge discovery in different data sources and with its representation through an ontologic structure and, together with ontology population, constitutes an approach for automating the

knowledge acquisition process. There are two fundamental aspects on ontology learning. The first one is the availability of prior knowledge. In other words, whether the learning process is performed from scratch or some prior knowledge is available; such prior knowledge is used in the construction of a first version of the ontology. Thus, a source of prior knowledge must demand little effort to be transformed into the first version of the ontology. This version is then extended automatically through learning procedures and manually by a knowledge engineer. The other aspect is the type of input used by the learning process. There are three different kinds of input: Structured data: database schemes, Semi-structured data: dictionaries like WordNet; and Unstructured data: natural language text documents, like the majority of the HTML based webpages;

III. Methods for Ontology Learning From Unstructured Texts

This section presents the most relevant methods and approaches used for ontology learning from text. The name of each method is the main reference in which the method or the approach has been described.

A. Aguirre and colleagues' method

This method [5] aims to enrich the concepts in existing large ontologies using text retrieved from the World Wide Web. The overall goal of this approach is to overcome two shortcomings of large ontologies like WordNet: the lack of topical links among concepts, and the proliferation of different senses for each concept. The method proposes four steps to enrich an existing ontology:

1. *Retrieve relevant documents for each concept.* The goal of this step is to retrieve documents related to an ontology concept from the web. The documents related to the same concept sense are grouped together to form collections, one for each sense.

2. Build topic signatures. The documents in each collection, related to a specific concept sense, have to be processed in order to extract the words and their frequencies using a statistical approach. Then, the data from one collection is compared with the data in the other collections. The words that have a distinctive frequency for one of the collections are grouped in a list, which then constitutes the topic signature for each concept sense.

3. Clustering word senses. Given a word, the concepts that lexicalize its word sense are hierarchically clustered. To carry out this task different topic signatures are compared to discover shared words, in order to determine overlaps between the signatures.

4. Evaluation. The topic signatures and hierarchical clusters are used to tag a given occurrence of a word in another corpus with the intended concept using different disambiguation algorithms.

B. Gupta and colleagues' approach

Gupta and colleagues [6] present an approach to acquire and maintain sublanguage WordNets from domain specific textual documents. The approach aims to enable rapid development of SubWordnets for Natural Language Processing (NLP) applications. The approach proposes an iterative three-step lexicon engineering cycle for developing SubWordNets as follows:

1. *Discover Concept Elements*: The goal of this step is to discover concept elements, which include words, generated multi-word phrases, and potential relationships among these elements that occur in input sublanguage documents. An unnamed relation between them could be discovered and suggested to the user. Subsequently, a user could identify the relation as of meronym/holonym type.

2. *Identify Concepts*: The objective of this step to identify new concepts and relations from phrases and relations discovered in the previous step. Concept identification is supported by grouping phrases into concept nodes and establishing concordance with synsets in WordNet. The new concept nodes and relationships can be used to update the SubWordNet.

3. *Maintain Concepts (Update SubWordNet)*: This step allows controlled insertion, deletion, and updating of concepts and relations derived from the previous step in a SubWordNet while maintaining its integrity.

Users can iterate through these steps with as many sublanguage documents as needed to develop SubWordNets and to maintain them on an ongoing basis.

C. Hwang's method

It focuses on the problems of locating, evaluating, retrieving, and merging information in an environment in which new information sources are constantly being added [7]. The procedure for generating the ontology has the following steps:

1. Human experts provide the system with a small number of *seed-words* that represent high-level concepts. Relevant documents will be collected from the web automatically.

2. The system *processes the incoming documents*, extracts only those phrases that contain seed-words, generates corresponding concept terms and places them in the "right" place in the ontology, and alerts the human experts of the changes. This feature is named "discover-and-alert". At the same time, it also collects candidates for seed-words for the next round of processing. The iteration continues a predefined number of times.

3. Several kinds of *relations* are extracted. Examples of relations are: "is-a", "part-of", "manufactured-by", "owned-by", etc, which are extracted based on linguistic features. The

"assoc-with" relation is used to define all relations that are not an "is-a" relation.
4. In each iteration, a *human expert is consulted* to ascertain the correctness of the concepts. If necessary, the expert has the right to make the correction and reconstruct the ontology.

D. Khan and Luo's method

This method [8] aims to build a domain ontology from text documents using clustering techniques and WordNet. The method constructs the ontology in a bottom-up fashion. The method proposes the following steps:
1. *Selection of the corpus to be used*. The user provides a selection of documents regarding to same domain.
2. *Hierarchy construction*. Using the set of documents provided in the previous sets, the method aims to create a set of clusters where each cluster may contain more than one document, and put them into the correct place in a hierarchy. Each node in this hierarchy is a cluster of documents.
3. *Concept assignment*. After building a hierarchy of clusters, a concept is assigned for each cluster in the hierarchy using a bottom-up fashion. Firstly, concepts associated with documents will be assigned to leaf nodes in the hierarchy. For each cluster of documents, it will be assigned a keyword, called topic that represents its content using a predefined topic categories. Then, this topic will be associated with an appropriate concept in WordNet. And finally, the interior node concepts will be assigned based on the concepts in the descendent nodes and their hyponyms in WordNet. The type of relation between concepts in the hierarchy is ignored; it is only possible to know that there is a relation between them.

E. Missikoff and colleagues' method

OntoLearn [9] is a method for ontology construction and enrichment using Natural Language (NL) and machine learning techniques. The method proposes using WordNet as a source of prior knowledge to build a core domain ontology, after pruning all of the unspecific domain concepts. The approaches followed by the method are: statistical, to determine the relevance of one term for the domain; and semantic interpretation, based on machine learning techniques, to identify the right sense of terms and the semantic relations among them.
The method proposes three main steps to achieve its goals
1. *Terminology extraction*. Terms and combinations of terms, such as "*last week*", are extracted from a parsed corpus using NL techniques.
2. *Semantic interpretation*. The main goals of this step are to determine the right concept sense for each component of a complex term, like a semantic disambiguation process, and then to identify the semantic relations holding among the concepts to build a complex concept. At the end of this step, a domain concept forest will be obtained, showing the taxonomic and other relationships among complex domain concepts represented by expressions.
3. *Creating the domain ontology*. This step aims to integrate the taxonomy obtained in the previous step with a core domain ontology. In the case that an existing domain ontology is not available, the method proposes to create a new one from WordNet, pruning concepts that are not related to the domain, and extending it with the new domain concept trees under the appropriate nodes.

F. Nobécourt approach

This work [10] presents an approach to build domain ontologies from texts using NLP techniques and a corpus. The method proposes two activities: modeling and representation.
1. The *modeling* activity includes a linguistic and a conceptual activity to find concepts and relationships between concepts.
2. The *representation activity* consists in the translation of the modeling schemata into an implementation language.

G. Roux and colleagues' approach

This work [11] aims to enrich an existing ontology with new concepts extracted from a parsed domain corpus using NL techniques. When a new word appears in the text that has not yet been referenced as a concept in the existing ontology, it is necessary to add this new word as a new concept. The approach is focused on managing new concepts with certain configurations of verbs (verb patterns) that will assign their position in the ontology. The verb patterns, used in this approach, are graphs in which one of the nodes is a verb that expects certain semantic attributes. These graphs will serve to connect the new term that matches in a specific semantic context, under corresponding nodes inside the ontology.

Table 1 summarizes the ontology learning methods from text.

TABLE 1. ONTOLOGY LEARNING METHODS FROM TEXT

Name	Goal	Techniques Used	Reuse Existing Ontologies	Evaluation Carried by
Aguirre and colleagues' method	To enrich concepts in existing ontologies	Statistical approach Clustering Topic signatures	Yes	User
Gupta and colleagues' approach	To build sublanguages in WordNet	NLP techniques Term-extraction techniques	Yes	Expert
Hwang's method	To elicit a taxonomy	NLP techniques ML techniques Statistical approach	No	Expert
Khan and Luo's method	To learn concepts	Clustering techniques Statistical approach	Yes	Expert
Missikoff and colleagues' method	To build taxonomies and to fuse with an existing ontology	NLP Statistical approach ML techniques	Yes	Expert
Nobécourt approach	To learn concept and relations among them	Linguistic analysis	No	User/expert
Roux and colleagues' approach	To enrich a taxonomy with new concepts	Verb-patterns	Yes	Expert

From the analysis presented above, we conclude:

- It does not exist a detailed methodology or method that guides the ontology learning process from text. There are methods that provide general guidelines.
- Most of these methods are mainly based on natural language analysis techniques, and use a corpus that guide the overall process.
- The most common ontology used by many methods is WordNet, which is used as initial ontology enriched with new concepts or relations.
- All these methods require the participation of an ontologist to evaluate the final ontology and the accuracy of the learning process.

IV. The Methods used for The Evaluation of Ontology Learning Approaches

Comparing techniques for learning ontologies is not a trivial task because for a given domain, there is not a unique possibility of conceptualization. There are two basic approaches for evaluating these systems: the evaluation of the underlying learning methods and the evaluation of the learned ontology. However, because of the difficulty concerning the measurement of the correctness of the learning procedure, the former approach is less addressed. The resulted ontologies can be compared by evaluating them in one of the following approaches [12]:

- Gold Standard evaluation: Comparing the ontology to a "golden standard". In a gold standard based ontology evaluation the quality of the ontology is expressed by its similarity to manually built gold standard ontology. A "golden standard" is a predefined ontology that is usually built manually from scratch by domain experts.
- Application based evaluation: Using the ontology in an application and evaluating the results. The ontology is evaluated by use it in some kind of application or task. Then the evaluation of the outputs of this application, or its performance on the given task will be used as evaluation for the used ontology.
- Data-driven evaluation: Comparisons with a source of data about the domain to be covered by the ontology. These are usually collections of text documents, web pages or dictionaries. This kind of evaluation is preferable in order to determine if the ontology refers to a particular topic of interest.
- Human evaluation: The evaluation is done by humans who try to assess how well the ontology meets a set of predefined criteria, standards, requirements, etc. It includes technical evaluation by the development team or by domain experts, and end users.

The establishment of formal, standard methods to evaluate ontology learning systems is still an open problem. There is no single best or preferred approach to ontology evaluation. The choice of a suitable approach must depend on the purpose of evaluation, the application in which the ontology is to be used, and on what aspect of the ontology that are being tried to evaluate.

V. Some Ontology Learning Tools From Text

This section presents some of the tools used for ontology learning.

A. CORPORUM-Ontobuilder

Its main aim is to be able to extract ontologies (mainly taxonomies) from natural language texts. CORPORUM-Ontobuilder extracts information from structured and unstructured documents using the tools named OntoWrappper [13] and OntoExtract [14]. Ontowrapper extracts information from on-line resources (e.g. names, email addresses, telephone numbers, etc.) and OntoExtract obtains taxonomies from natural language texts.

B. DOE: Differential Ontology Editor

DOE [15] is a simple ontology editor that allows the user to build ontologies in three steps. In the first step, the user develops taxonomies of concept and relation by justifying explicitly their position in the hierarchy. In a second step, the taxonomies are imported and the user can add constraints onto the domains of the relations. Finally, in a third step, the ontology is translated into a KR language.

C. KEA: Keyphrases Extraction Algorithm

KEA [16] automatically extracts keyphrases from the full text of documents. The extraction algorithm has two stages: *Training stage* that uses a set of training documents for which the author's keyphrases are known. For each training document, candidate phrases are identified and different features values are calculated. To reduce the size of the training set, any phrase that occurs only once in the document is discarded. Each phrase is then marked as a keyphrase or a non-keyphrase, using the actual keyphrases for that document. *Extraction stage*. To select keyphrases from a new document, KEA determines candidate phrases and feature values, and then applies the model built during the training stage. The model determines the overall probability that each candidate is a keyphrase, and then a post-processing operation selects the best set of keyphrases.

D. LTG (Language Technology Group) Text Processing Workbench

It is a set of computational tools for uncovering internal structure in natural language texts written in English. The main idea behind the workbench is the independence of the text representation and text analysis [17]. In LTG, ontology learning is performed in two sequential steps: representation and analysis. At the representation step, the text is converted from a sequence of characters to features of interest by means

of annotation tools. At the analysis step, those features are used by statistics-gathering tools and inference tools for finding significant correlations in the texts. The workbench uses methods from computational linguistics, information retrieval and knowledge engineering.

E. OntoLearn Tool

It aims to extract relevant domain terms from a corpus of text, relate them to appropriate concepts in a general-purpose ontology, and to detect relations among the concepts [18]. OntoLearn extracts terminology from a corpus of domain text, such as specialized Web sites. The system then filters the terms using natural language processing and statistical techniques that perform comparative analysis across different domains, or contrasting corpora. This analysis identifies terminology that is used in the target domain but is not seen in other domains. Next, it uses the WordNet and SemCor lexical knowledge bases to perform semantic interpretation of the terms. The tool then relates concepts according to taxonomic (kind-of) and other semantic relations, generating a domain concept forest. For this purpose, WordNet and a rule-based inductive-learning method have been used to extract such relations. Finally, OntoLearn integrates the domain concept forest with WordNet to create a pruned and specialized view of the domain ontology. The validation of the process is performed by an expert.

F. SOAT: a Semi-Automatic Domain Ontology Acquisition Tool

It allows a semi-automatic domain ontology acquisition from a domain corpus. The main objective of the tool is to extract relationships from parsed sentences based on applying phrase-rules to identify keywords with strong semantic links like hyperonym or synonym [19]. The acquisition process is based on using InfoMap, a knowledge representation framework, that integrates linguistic, commonsense, and domain knowledge. InfoMap has been developed to perform natural language understanding, and to capture the topic words, usually pairs of noun and verb, or noun and noun in a sentence. InfoMap has two major relations between concepts: taxonomic relations (category and synonym) and non-taxonomic (attribute and event).

G. TFIDF-based term classification system

It aims to detect relevant domain terms and to learn relations that hold among them [20]. The tool has the following three main components. A *based single-word term classifier* that is used to extract single words from a corpus. A *lexico-syntactical pattern finder*, that has two sub-modules: the first one is for learning patterns based on the set of known relations (using GermaNet or WordNet) and implements the interfaces needed to interoperate with these two systems; the second is for learning patterns based on term co-location methods. The final component is the *relation extractor*.

H. TERMINAE

It integrates linguistic tools and knowledge engineering tools [21]. The linguistic tool allows defining terminological forms from the analysis of term occurrences in a corpus. The ontologists analyse the uses of the term in the corpus to define the meanings of the terms. The knowledge engineering tool involves an editor and a browser for the ontology. The tool helps to represent terminological forms as a concept (called terminological concept).

I. TextStorm and Clouds

It semi-automatically constructs a semantic network which contains only concepts and their relations, using a relevant text for the target domain [22]. It is composed of two main modules: TextStorm [23] and Clouds [24] that perform complementary activities to construct a semantic network. TextStorm deals with the task of extracting relations between concepts from a text file using NL techniques, while Clouds is concentrated on completing these relations and extrapolating rules about the knowledge previously extracted using NL techniques.

Table 2 summarizes the ontology learning tools from text.

TABLE 2. ONTOLOGY LEARNING TOOLS FROM TEXT

Name	Goal	Learning Technique	Interoperability	User Intervention
CORPORUM -Ontobuilder	To extract initial taxonomy	Linguistic and semantic techniques	OntoWrapper and OntoExtract	Not necessary
DOE	To help to the ontologist in the process of building an ontology	Differential Semantic	None	Whole process
KEA	To Summarize documents extracting keywords	Statistical approach ML techniques Lexical processing	WEKA ML Workbench	Evaluation
LTG	To discover internal relations of texts in NL	Statistic Inference Linguistic technique	Can be used to perform the knowledge acquisition to any other ontology development tool	Whole process
OntoLearn Tool	To enrich a domain ontology	NLP techniques ML techniques	None	Evaluation
SOAT	Acquisition of relationships	Phrase-patterns	Information not available	Information not available
TFIDF-based term classification system	To learn concepts and relation between them	Text-mining Statistical approach	SPPC NLP tool	Evaluation
TERMINAE	To build an initial ontology	Conceptual clustering	Information not Available	Validation
TextStorm and Clouds	To build a taxonomy	NLP techniques Linguistic hypothesis	Information not available	Whole process

From the analysis presented above, we conclude:

- Most of these tools perform NLP to extract linguistic and semantic knowledge from the corpus used for learning.

- It does not exist a fully automatic tool that carries out the learning process.
- There is neither tool that evaluates the accuracy of the learning process nor to compare different results obtained using different learning techniques.

VI. CONCLUSIONS

Ontologies are the vehicle by which we can model and share the knowledge among various applications in a specific domain. This work presented a survey and a definition of the set of tasks known as ontology learning. The tasks involved in the ontology learning from text were identified and the most commonly cited approaches for this problem were introduced. Also the paper presents the tools used for ontology learning from text. The field of ontology learning builds upon well founded methods from knowledge acquisition, machine learning and natural language processing. Although progress has been made over the last years, this field of research has not yet reached the goal of fully automating the ontology development process. Another great open question related to ontology learning is the evaluation of the proposed methods. As shown in this paper there is already some work conducted in this direction but the establishment of formal, standard methods to evaluate ontology learning systems by proving their learning methods or proving the accuracy, efficiency and completeness of the built ontology is still an open problem. Because of the complexity of ontology development and the importance of the ontologies for the Semantic Web (given that the ontologies constitute its backbone), we believe that ontology learning will remain an active and central field of research, at least, over the next years.

REFERENCES

[1] T. Gruber. Towards principles for the design of ontologies used for knowledge sharing. Int. J. of Human and Computer Studies, 43:907–928, 1994.

[2] Y. Ding. 'Ontology: the Enabler for the Semantic Web'. Journal of Information Science. 27 (6), 2001.

[3] Buitelaar, P., Cimiano, P., Magnini, B.: Ontology learning from text: An overview. ontology learning from text: Methods, evaluation and applications. Frontiers in Artificial Intelligence and Applications Series 123, 2005.

[4] Cimiano, P., V¨olker, J., Studer, R.: Ontologies on demand? - a description of the state-of-the-art, applications, challenges and trends for ontology learning from text. Information, Wissenschaft und Praxis 57(6-7) PP: 315–320, 2006.

[5] Agirre, E., Ansa, O., Hovy, E., and Martinez, D. Enriching very large ontologies using the WWW. In Proceedings of the Workshop on Ontology Construction of the European Conference of AI (ECAI-00), 2000.

[6] Gupta, K.M., Aha, D.W., Marsh, E., and Maney, T. Architecture for engineering sublanguage WordNets. In Proceedings of the First International Conference on Global WordNet (pp. 207-215). Mysore, India: Central Institute of Indian Languages, 2002.

[7] Hwang, C. H. Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. In. Proceedings of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99), Linköping, Sweden, July 29-30, 1999.

[8] Khan L., and Luo F. Ontology Construction for Information Selection In Proc. of 14th IEEE International Conference on Tools with Artificial Intelligence, pp. 122-127, Washington DC, November 2002.

[9] Missikoff M., Navigli R., and Velardi P. The Usable Ontology: An Environment for Building and Assessing a Domain Ontology Research paper at International Semantic Web Conference (ISWC), Sardinia, Italia, June 9-12th, 2002.

[10] Nobécourt J. A method to build formal ontologies from text. In: EKAW-2000 Workshop on ontologies and text, Juan-Les-Pins, France, 2000.

[11] Roux C., Proux D., Rechermann F., and Julliard L. An ontology enrichment method for a pragmatic information extraction system gathering data on genetic interactions. Position paper in Proceedings of the ECAI2000 Workshop on Ontology Learning (OL2000), Berlin, Germany. August 2000.

[12] Brank, J., Grobelnik, M., and Mladenic, D. A survey of ontology evaluation techniques. In Proceedings of the 8th Int. multi-conference Information Society IS-2005.

[13] Engels R. CORPORUM-OntoWrapper. Extraction of structured information from web based resources. Deliverable 7 – Ontoknowledge. http://www.ontonowledge.org/del.shtml, 2001.

[14] Engels R. CORPORUM-OntoExtract. Ontology Extraction Tool. Deliverable 6 Ontoknowledge. http://www.ontonowledge.org/del.shtml, 2001.

[15] Asunción Gómez-Pérez, David Manzano-Macho, Deliverable 1.5: A survey of ontology learning methods and techniques, This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2000-29243, 2003.

[16] Jones, S. and Paynter, G.W. Automatic extraction of document keyphrases for use in digital libraries: evaluation and applications. Journal of the American Society for Information Science and Technology (JASIST), 2002.

[17] Mikheev, A. Finch, S.(1997) A Workbench for Finding Structure in Texts. Proceedings of ANLP-97 (Washington D.C.). ACL March 1997.

[18] Velardi P., Navigli R., and Missikoff M. Integrated approach for Web ontology learning and engineering. IEEE Computer - November 2002.

[19] Wu S.H, Hsu W.L. SOAT: A Semi-Automatic Domain Ontology Acquisition Tool from Chinese Corpus. In the 19th International Conference on Computational Linguistics, Howard International House and Academia Sinica, Taipei, Taiwan, 2002.

[20] Xu F., Kurz D., Piskorski J., and Schmeier S. A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Bootstrapping. In Proceedings of LREC 2002, the third international conference on language resources and evaluation, Las Palmas, Canary island, Spain, May 2002.

[21] Biébow B, Szulman S. TERMINAE: a linguistic-based tool for the building of a domain ontology. In EKAW'99 Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling and management. Dagstuhl, Germany, LCNS, pages 49-66, Berlin, 1999. Springer-Verlag, 1999.

[22] Pereira, F. C. Modelling Divergent Production: a multi domain Approach. European Conference of Artificial Intelligence, ECAI'98, Brighton, UK, 1998.

[23] Oliveira A., Pereira F.C., and Cardoso A. Automatic Reading and Learning from Text. In Proceedings of International Symposium on Artificial Intelligence, ISAI'2001. December, 2001.

[24] Pereira, F. C.; Oliveira, A. and Cardoso, A. Extracting Concept Maps with Clouds. Argentine Symposium of Artificial Intelligence (ASAI 2000), Buenos Aires, Argentina, 2000.

Comparative study of Intelligent Classification techniques for Brain Magnetic Resonance Imaging

Heba Mohsen, Abdel-Badeeh M. Salem
Faculty of Computer and Information Science
Ain Shams University, Abbassia
Cairo, Egypt

El-Sayed Ahmed El-Dahshan
Faculty of Science
Ain Shams University, Abbassia
Cairo, Egypt

***Abstract*—Brain tissue classification from Magnetic Resonance Imaging (MRI) is of great importance for research and clinical studies of the normal and diseased human brain. All MRI classification methods are sensitive to overlap in the tissue intensity distributions. Such overlaps are caused by inherent limitations of the image acquisition process, such as noise, intensity non-uniformity, and partial volume effect. Several approaches have been proposed to address this limitation of intensity-based classification. The objective of this paper is to make a comparative study on the recent published classification techniques for the brain magnetic resonance images (MRI). The contribution of this study is to determine the advantages and disadvantages of each technique and develop robust classification technique capable to perform an efficient and automated MRI normal/abnormal brain images classification.**

Keywords: Brain Magnetic Resonance Imaging, Feature Extractions, Classification, Intelligent Systems, Medical Informatics.

I. INTRODUCTION

Brain is the kernel part of the body. Brain has a very complex structure. Brain is hidden from direct view by the protective skull. This skull gives brain protection from injuries as well as it hinders the study of its function in both health and disease. But brain can be affected by a problem which cause change in its normal structure and its normal behavior .This problem is known as brain tumor [1]. A brain tumor is any mass that cause an abnormal growth of cells within the brain or inside the skull, which can be cancerous (malign) or non-cancerous (benign). It is defined as any intracranial tumor created by abnormal and uncontrolled cell division. This type of brain tumor constitutes one of the most frequent causes of death among the human being in the world. Brain tumor effects may not be the same for each person, and they may even change from one treatment session to the next. Detection of tumor in the earliest stage is the key for its successful treatment [2, 3].

The use of computer technology in medical decision support is now widespread and pervasive across a wide range of medical area, such as cancer research, gastroenterology, heart diseases, and brain tumors. To this need, image analysis techniques have been employed in previous studies for the extraction of diagnostic information from MRI. These studies have employed pattern recognition and texture analysis techniques to characterize human brain tumors. Fully automatic normal and diseased human brain classification can be obtained from magnetic resonance images; which is a great importance for research and clinical studies [4, 5].

As brain tumors can have a variety of shapes and sizes; it can appear at any location and in different image intensities. The main aim of this study is to classify the brain images to normal and abnormal. Image classification is an important step for quantitative analysis in order to detect pathology or quantify disease response to therapy. Image classification is a process to partition an image into a set of distinct classes with uniform or homogeneous attributes such as textures or intensity. For classification of the images different features of the image are extracted. These features are used for classifying the brain MR image as normal and abnormal [1, 6].

Methods for fully automatic brain tissue classification typically rely on an existing anatomical model for localizing a training set for each tissue class to be labeled, e.g. gray matter, white matter, CSF (cerebro-spinal fluid) and Abnormal brain tissues. This assumption of normal anatomical distribution of tissue types makes them sensitive to any deviations from the model due to pathology, or simply due to normal anatomical variability between individuals. Also, there may be situations when the only model available was constructed from a different human population than the image to be classified [7].

All MRI classification methods are sensitive to overlap in the tissue intensity distributions [7]. Such overlaps are caused by inherent limitations of the image acquisition process, such as noise, intensity non-uniformity (INU, also known as bias field), and partial volume effect (as a consequence of the finite resolution of the imaging process, the image voxels may contain a mixture of more than one tissue type, which all

contribute to the measured signal). Several approaches [8] have been proposed to address this limitation of intensity-based classification. Researchers used conventional MR imaging and echo-planar relative cerebral blood volume (rCBV) maps calculated from perfusion imaging to differentiate between high grade and low grade neoplasms, or assessed the contribution of MR perfusion alone in differentiating certain tumor types.

II. BRAIN IMAGING TECHNIQUES

Brain imaging techniques allow doctors and researchers to view activity or problems within the human brain, without invasive neurosurgery. There are a number of accepted, safe imaging techniques in use today in research facilities and hospitals throughout the world.The cells which supplies the brain in the arteries are tightly bound together thereby routine laboratory test are inadequate to analyze the chemistry of brain. There are many imaging modalities that allow the doctors and researchers to study the brain by looking at the brain non-invasively [1].

Recently, imaging techniques, like are used to locate the position and extent of brain tumors [5]. MRI can provide information about brain tissues, from a variety of excitation sequences. Compared with other diagnostic imaging modalities, such as computerized tomography (CT), MRI provides superior contrast for different brain tissues. Additionally, MR images encapsulate valuable information regarding numerous tissue parameters (proton density, spin–lattice (T1) and spin–spin (T2) relaxation times, flow velocity and chemical shift), which lead to more accurate brain tissue characterization. These unique advantages have characterized MRI as the method of choice in brain tumor studies.

MRI is often the medical imaging method of choice when soft tissue delineation is necessary. This is especially true for any attempt to classify brain tissues. Radiologist used it for the visualization of the internal structure of the body. MRI provides rich information about human soft tissues anatomy. MRI helps for diagnosis of the brain tumor. Images obtained by the MRI are used for analyzing and studying the behavior of the brain. Image intensity in MRI depends upon four parameters. One is proton density (PD) which is determined by the relative concentration of water molecules. Other three parameters are T1, T2, and T2* relaxation, which reflect different features of the local environment of individual protons. [1, 4].

An attractive feature of MRI is that different contrasts between tissue types (multispectral image data of the same subject) can be easily obtained. For that in the recent years, MRI has evolved into a popular technique to study the human brain. This non-invasive technique can provide high resolution spatial images and its rich information content can be suitably utilized to develop automated diagnostic tools, which can aid the medical fraternity to draw quicker and easier inferences about the condition of the brain under study.

III. GENERAL METHODOLOGY FOR AN AUTOMATED MRI SYSTEM

The development of automated tools can be of immense importance to help in diagnosis, prognosis and presurgical and post-surgical procedures, depending on whether the subject is a healthy one or is a pathological subject, suffering from some brain disorder, e.g. Alzheimer's disease, Parkinson's disease etc. The extraordinary level of detail that can be obtained with brain MR images can be efficiently utilized by performing some powerful signal or image processing techniques, especially suitable for automated analysis. This is because, with the huge information repository associated with MRIs, it becomes almost impossible to manually interpret each image, necessitating the development of automated tools [9].

An automated MRI system consists of multiple phases. Fig.1 shows the details of the system. First MR image for diagnosis is provided to the system as an input. Second step of the proposed system is to extract features from this input image. After feature extraction, these features independently are used for classification as malignant and benign MR image. It classifies the brain image on the bases of multiple classifiers. No more processing is required once the MR image is determined as benign. But when the MR image is determined as malignant by the classifier it is further processed for extracting tumor portion from it [1].

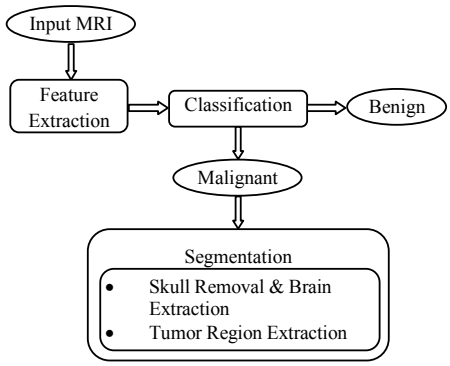


Figure 1. General Methodology for a MRI automated system

a) *Feature Extraction:* The transformation of an image into its set of features is known as feature extraction.

Useful features of the image are extracted from the image for classification purpose. It is a challenging task to extract good feature set for classification. There are many techniques for feature extraction e.g. texture Features, gabor features, feature based on wavelet transform, principal component analysis, minimum noise fraction transform, discriminant analysis, decision boundary feature extraction, non-parametric weighted feature extraction and spectral mixture analysis[1].

b) *Classification:* Classification is the technique for classifying the input patterns into analogous classes. Selection of a suitable classifier requires consideration of many factors: (a) Classification accuracy, (b) Algorithm performance, (c) Computational resources.

c) *Segmentation:* After the classification phase segmentation of the malignant images is performed into two steps: Skull removal and brain region extraction (detection) and Brain Tumor Extraction.

IV. COMPARATIVE STUDY OF CLASSIFICATION TECHNIQUES FOR BRAIN MRI

In this study we present most recent published classification techniques. Table 1 shows a comparison between 12 recent classification techniques for Brain MRI that published during 2006-2011. It illustrate that there are two basic types of classification. One is known as unsupervised classification and other is known as supervised classification. Supervised classification methods depend on the examples of information classes in the images and derive a prior model or parameters from the training sets. Unsupervised methods examine the images without specific training data and divide the image pixels into groups presented in the pixel values according to classification criteria. Unsupervised methods have the advantage of being fast and not requiring training information which is sometimes unavailable [6].

TABLE 1. CLASSIFICATION TECHNIQUES FOR BRAIN MRI

Authors	Feature Extraction	Classification Technique	Accuracy
Herlidou-Meme et al. (2003) [10]	Texture Analysis	Hierarchical Ascending Classification	
Maitra et al. (2007) [9]	improved orthogonal discrete wavelet transform (DWT)	fuzzy c-means (FCM) clustering	100%
Georgiadis et al. (2007) [5]	Texture feature extraction + non-parametric Wilcoxon rank sum test method[15]	Non linear least squares features transformation with probabilistic neural networks (LSFT-PNN)	94.03%
Wang et al. (2008) [6]	-----	Multiscale fuzzy C-means (MsFCM)	88%
El-Dahshan et al. (2010) [4]	discrete wavelet transformation (DWT) + principal component analysis (PCA)	• feed forward backpropagation artificial neural network (FP-ANN) • k-nearest neighbor (k-NN)	97% 98.6%
Latif et al. (2010) [1]	Texture feature extraction	ensemble base classifier	99%
Zacharaki et al. (2009) [8]	Manual feature extraction + t-test / Constrained Linear Discriminant Analysis (CLDA)	• Linear Discriminant Analysis (LDA) • k-nearest neighbor (k-NN) • Support Vector Machine (SVM)	81-85% 89% 91%
Kharrat et al. (2010) [3]	Spatial gray level dependence method (SGLDM) Wavelet transform (WT) + Spatial gray level dependence method (SGLDM)	• Genetic Algorithm (GA) + Support Vector Machine (SVM) • Genetic Algorithm (GA) + Support Vector Machine (SVM)	95% 97%
Zhang et al. (2010) [11]	discrete wavelet transformation (DWT) + principal component analysis (PCA)	adaptive chaotic particle swarm optimization - forward neural network (ACPSO - FNN)	98.75%
Selvaraj et al. (2007) [12]	Texture feature extraction	• Least Squares Support Vector Machines (LS-SVM) • Support Vector Machine (SVM) • Multi Layer Perceptron (MLP) • Radial Basis Function (RBF) • k-nearest neighbor (k-NN)	97% 94% 91.5% 92% 89.65%
Chaplot et al. (2006) [13]	Wavelet transform (Daubechies-4 wavelet)	• Self-Organizing maps (SOM) • Support Vector Machine (SVM) • Support Vector Machine (SVM) with radial basis function based kernel	94% 96% 98%
Zhang et al. (2011) [14]	Wavelet transformation + principal component analysis (PCA)	Back Propagation neural network (BPNN)	100%

Recent works have shown that classification of human brain in magnetic resonance images (MRI) is possible via supervised techniques such as artificial neural networks and support vector machine (SVM), and unsupervised classification techniques such as self-organization map (SOM) and fuzzy c-means. Other supervised classification techniques, such as k-nearest neighbors (k-NN) can be used to classify the normal/ pathological T2-weighted MRI images [4].

From Table 1 it can be seen that :

1. The most common methods for feature extraction are discrete wavelet transform (DWT) and Texture analysis.
2. Most common methods for classification are fuzzy c-means that gives best accuracy combined with a pre-feature extraction and different neural network techniques.
3. Fuzzy c-means clustering, ensemble , k-nearest neighbor and feed forward ANN) techniques gives very high accuracy (in the range 97% - 100%).

V. CONCLUSION

There are many proposed systems developed for the aim of diagnosing the brain tumor from Brian MR Images. As MRI provides rich information about human soft tissues anatomy that can be suitably utilized to develop automated diagnostic tools, which can aid the medical fraternity to draw quicker and easier inferences about the condition of the brain under study. These systems have multiple phases to perform the diagnoses. Most important phases are feature extraction to extract features that to be used then in classification phase. In this study we present a comparative study for different approaches of Brian tumors Classification in proposed diagnosing systems. This area of research still has a lot to cover. We stated the proposed systems according to feature extraction methods and classification approaches along with their accuracy achieved in a comparative study that could be used to develop scalable systems that will be applicable in real life scenarios.

VI. REFERENCES

[1] G. Latif, S. B. Kazmi, M. A. Jaffar, A. M. Mirza, Classification and Segmentation of Brain Tumor using Texture Analysis, Proceedings of the 9th Wseas International Conference on Artificial Intelligence Knowledge Engineering and Data Bases, pp. 147-155, 2010

[2] P. Rajendran, M. Madheswaran, Hybrid Medical Image Classification Using Association Rule Mining with Decision Tree Algorithm, Journal of Computing vol. 2, pp. 127-136, 2010

[3] A. Kharrat, K. Gasmi, M.B. Messaoud, N. Benamrane, M. Abid, A Hybrid Approach for Automatic Classification of Brain MRI Using Genetic Algorithm and Support Vector Machine, Leonardo Journal of Sciences, vol. 9, pp. 71-82, 2010

[4] E.S. El-Dahshan, T. Hosny, A.B. M. Salem, Hybrid intelligent techniques for MRI brain images classification, Digital Signal Processing, vol. 20, pp. 433-441, 2010

[5] P. Georgiadis, D. Cavouras, I. Kalatzis, A. Daskalakis, G.C. Kagadis, K. Sifaki, M. Malamas, G. Nikiforidis, E. Solomou, Improving brain tumor characterization on MRI by probabilistic neural networks and non-linear transformation of textural features, Comput Methods Programs Biomed vol. 89, pp. 24-32, 2008

[6] H. Wang, B. Fei, A modified fuzzy C-means classification method using a multiscale diffusion filtering scheme, Medical Image Analysis, vol. 13, pp. 193–202, 2009

[7] C.A. Cocosco, A.P. Zijdenbos, A.C. Evans, A fully automatic and robust brain MRI tissue classification method, Medical Image Analysis, vol. 7, pp. 513–527, 2003

[8] E.I. Zacharaki, S. Wang, S. Chawla, D. Soo Yoo, R. Wolf, E.R. Melhem, C. Davatzikos, Classification of brain tumor type and grade using MRI texture and shape in a machine learning scheme, Magn Reson Med., vol. 6, pp. 1609–1618, 2009

[9] M. Maitra, A. Chatterjee, Hybrid multiresolution Slantlet transform and fuzzy c-means clustering approach for normal-pathological brain MR image segregation, Medical Engineering & Physics, vol. 30, pp. 615–623, 2008

[10] S. Herlidou-Meme, J.M. Constansb, B. Carsinc, D. Oliviea,c, P.A. Eliata, L. Nadal-Desbaratse, C. Gondryf, E. Le Rumeura, I. Idy-Perettie, J.D. de Certainesa, MRI texture analysis on texture test objects, normal brain and intracranial tumors, Magnetic Resonance Imaging, vol. 21, pp. 989–993, 2003

[11] Y. Zhang, S. Wang, L. Wu, A Novel Method for Magnetic Resonance Brain Image Classification Based on Adaptive Chaotic PSO, Progress In Electromagnetics Research, vol. 109, pp. 325-343, 2010

[12] H. Selvaraj, S. Thamarai Selvi, D. Selvathi, L. Gewali, Brain MRI Slices Classification Using Least Squares Support Vector Machine, IC-MED, vol. 1, pp. 21-33, 2007.

[13] S. Chaplot, L.M. Patnaik, N.R. Jagannathan, Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network, Biomedical Signal Processing and Control, vol. 1, pp. 86–92, 2006

[14] Y. Zhang, Z. Dong, L. Wu, S. Wang, A hybrid method for MRI brain image classification, Expert Systems with Applications, vol. 38, pp. 10049–10053, 2011

[15] K.Q.Weinberger, L.K. Saul, Distance Metric Learning for Large Margin Nearest Neighbor Classification, Journal of Machine Learning Research, vol. 10, pp. 207-244, 2009

[16] B.H.ChandraShekar, Dr.G.Shoba, Classification Of Documents Using Kohonen's Self-Organizing Map, IJCTE, vol. 5, pp. 610-613, 2009

[17] O. Taylor, J. Tait, J. Maclntyre, Improved Classification for a Data Fusing Kohonen Self Organizing Map Using a Dynamic Thresholding Technique, In IJCAI, pp. 828-832, 1999

[18] CBTRUS (2011). CBTRUS Statistical Report: Primary Brain and Central Nervous System Tumors Diagnosed in the United States in 2004-2007. Source: Central Brain Tumor Registry of the United States, Hinsdale, IL. website: www.cbtrus.org

[19] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics, vol. 1, pp. 80–83, 1945

[20] K.R. Porter, B.J. McCarthy, S. Freels, Y. Kim, F.G. Davis, Prevalence estimates for primary brain tumors in the US by age, gender, behavior, and histology, Neuro–Oncology, vol. 12, pp. 520-527, 2010

[21] M.H.F. Zarandi, M. Zarinbal, and M. Izadi, Systematic image processing for diagnosing brain tumors: A Type-II fuzzy expert system approach, Appl. Soft Comput., pp. 285-294, 2011

Generalization and Specialization Using Extended Conceptual Graphs

Erika Baksa-Varga
Department of Information Technology
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
Email: vargae@iit.uni-miskolc.hu

László Kovács
Department of Information Technology
University of Miskolc
H-3515 Miskolc-Egyetemváros, Hungary
Email: kovacs@iit.uni-miskolc.hu

***Abstract*—The final goal of our research is to show that the performance of statistical rule induction can be improved by augmenting training data with semantic information. In order to prove this hypothesis, a statistical grammar induction system is to be created the knowledge base of which is represented by Extended Conceptual Graphs (ECGs). Since generalization and specialization are the basic operations of induction, they are of great significance in machine learning. Generalization using ECG graphs can be interpreted in several levels, amongst which concept generalization has been implemented and defined as the process by which abstract ECG concepts are derived from lower-level concepts. For the accomplishment of this operation a domain-specific ECG element instance type lattice $(T', \prec)$ has been generated for the given test environment. The definition of the $\prec$ relation on element instances can be extended to a partial relation $\preceq$ on ECG diagram graphs, according to which $\Gamma_1 \preceq \Gamma_2$ if graph Γ_1 is more 'specialized' than Γ_2. The paper aims at investigating the least common generalized graph and the greatest common specialized graph of two ECG graphs.**

Index Terms*— semantic modeling, knowledge representation, machine learning, conceptualization.

I. Introduction

The main motivation for the research is to develop a new general rule learning methodology that alloys statistics with semantics. The actual learning problem is chosen to be grammar induction, because symbolic languages have a fairly complex systems of rules (grammars), so they must be considered when developing a general methodology. Also, grammar induction has many application areas, such as computational linguistics, chemistry or pattern matching. The first phase of the research has covered the specification of an appropriate semantic knowledge representation model (called Extended Conceptual Graph, ECG model [1]) optimized for grammar induction, which is used for representing the knowledge base of the agent examined. The capabilities of the grammar learning agent are fixed in advance, which are

1) pattern recognition: the ability to recognize the objects of its direct environment and their relations;
2) association: the ability of relating pieces of information based on its stored knowledge; and
3) generalization: the ability of introducing new abstract concepts by extracting the common characteristics of existing knowledge items.

These operations define the process of conceptualization within the grammar learning agent at the end of which stands the generalized knowledge an agent can obtain from the samples observed. This can be given as a generalized accumulated ECG diagram graph built up from a set of primary-level ECG diagram graphs through several generalization stages. On the other hand, the specialization of two general ECG diagram graphs is formulated as their greatest common specialization, which can be defined as the maximal common restriction of the two graphs.

The paper is aimed at demonstrating both processes of generalization and specialization. Accordingly, the paper is organized as follows. Section II introduces a related paper that gave the guidelines for the present investigations. Section III gives a brief introduction to the ECG semantic model. Then Section IV gives the details of the process of conceptualization using ECG graphs, involving the operations of graph matching and association (IV-B) and generalization (IV-C). This section also specifies the ECG element type lattices (IV-A) required for modeling conceptualization and an example environment (IV-D). After these, Section V defines the process of specialization and demonstrates it through an example. Finally, the paper ends up with a conclusion (Section VI).

II. Related Work

[2] outlines the principles of a Prolog-like resolution method which allows for expressing a large amount of background knowledge in terms of Sowa's conceptual graphs and performing deduction on very large linguistic and semantic domains. The paper introduces two algorithms: one for generating the greatest common specialized graph of two conceptual graphs (*maximal join of two graphs*), and one for generating the least common generalized graph that can be obtained from two conceptual graphs (*generalization of two graphs*). This CG processor is the main component of the KALIPSOS general system for knowledge acquisition from texts that is developed at IBM Paris Scientific Center. The success of the project has motivated the effort of simulating the process of generalization and specialization using ECG diagram graphs.

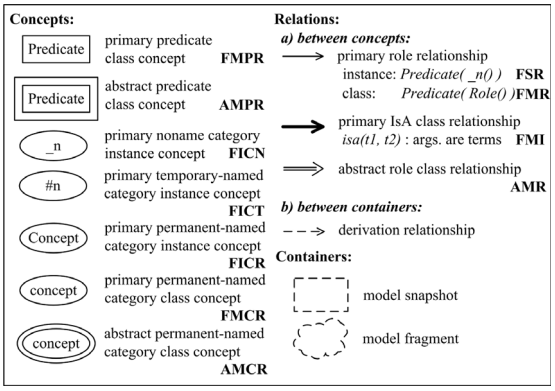


Fig. 1. Graphical components of ECG diagram

III. Modeling with Extended Conceptual Graphs

The Extended Conceptual Graph (ECG) model is a semantic modeling language which can be given in two equivalent forms: in an adequately extended higher-order predicate logic based textual format (ECG-HOPL [3]) and in a graphical representation called ECG diagram (see its components in Fig. 1). It is computationally tractable while highly expressive, that is it covers a wide range of linguistic phenomena. The model is designed for knowledge representation in learning agents and is specifically optimized for grammar induction. The capabilities of the agents are fixed in advance, and they are defined so that they are able to detect objects in the environment, their attributes, and the relationships between them; where the set of recognizable attributes and relationships are pre-defined. The main characteristics of the model can be summarized as follows.

a) Main building blocks of the model: The main building blocks of the model are concepts, relationships, and containers which serve for structuring the model. The 'world' is built up of interconnected ECG model fragments representing separate observations, containing exactly one kernel predicate and having 'true' truth value. Since the model bears the features of ontologies, it can be considered as an *ontology modeling language*.

b) Predicate-centeredness: In contrast with other existing semantic models – which have been deeply examined in view of their expressive power in [4] – ECG is predicate-centered[1]. That is predicate concepts, which are distinguished from non-predicate concepts, are the kernels of atomic propositions. In the center of an ECG model fragment stays the kernel predicate, and each basic ECG graphical structure is organized around a predicate.

c) Multiple conceptualization levels: In the ECG model, the process of conceptualization occurs at two levels. The primary level of the ECG model serves for the direct mapping of environment objects and relationships into primary-level knowledge items. At the abstract level temporal and other complex relationship types are also managed.

d) Distinction between apriori and learned elements: ECG differentiates between several categories of concept and relationship types both at the primary and at the abstract level.

e) Flexibility: The ECG model is able to grasp the semantic content of situations. The elements of the environment can be represented by the relatively small, fixed set of ECG model elements. This means that several environment elements are mapped to the same ECG model element, which has therefore flexible semantic assignment.

f) Extendibility: The ECG model is a recursive, compositional system: that is infinitely many statements can be constructed from the small finite set of model elements.

[1]Sowa's Conceptual Graph (CG) model [5] is also predicate-centered, but there are no other connections between the two models.

IV. The Process of Conceptualization

In terms of machine learning, concept formation is the process by which an agent learns to sort specific experiences into general rules or classes. In order to make learning feasible in complex domains, abstraction and generalization operators are often applied to make the problem tractable [6]. In the present approach, the learning agent builds up its knowledge base by

1) incorporating and relating the information elements observed – which are instance-level ontologies represented by ECG diagram graphs – to its existing (and continuously evolving) knowledge base (*association*); and
2) introducing new (not observed) higher-level concepts into the knowledge base, thereby reducing its complexity (*generalization*).

The higher-level concepts to be introduced are pre-defined in a domain-specific concept lattice which is used for representing concept generalization structures [7]. Its generation from the given domain is called *abstraction*. A widely accepted formalism for conceptualization is the theory of Formal Concept Analysis (FCA) [8]. In FCA, there are two main variants of concept set building algorithms. The methods of the first group work in batch mode, assuming that every element of the context table is already present before starting the concept lattice building. The other group of proposals uses an incremental lattice building method [9].

Our concept lattice building algorithm belongs to the first group. It uses the information present in the samples, and also the abstract element types defined by the ECG model. There are two abstract concept types which can be used in generalization, that is

- AMCR: abstract category concept, for generalizing concepts in T_{cc};
- AMPR: abstract predicate concept, for generalizing concepts in T_{pc}.

However, they can not be generated automatically from the samples. They need to be manually added to the lattice.

A. ECG Element Type Lattices

In the ECG model, concepts and relations (elements) are given by two attributes: a type and a caption, in the form of

TABLE I
CLASSIFICATION OF ECG ELEMENT CATEGORY TYPES

Subset of T	**Element category types**
T_{cc}: category concept types	FICN, FICT, FICR, FMCR, AMCR
T_{pc}: predicate concept types	FMPR, AMPR
T_{rr}: semantic role relation types	FSR, FMR, AMR
T_{sr}: specialization relation type	FMI

$type : caption$. Formally, each $ec \in EC$ element category in the ECG model is given as $ec = type_c : caption$, where $type_c \in T$ and $caption$ is a string representing the name of the corresponding element category. The T set of ECG element category types is the union of the subsets listed in Table I.

ECG element category types can be merged in a lattice $(T, <)$, whose partial ordering relation $<$ can be interpreted as a categorical generalization relation. The top and the bottom element categories of the $(T, <)$ lattice are UNIV (the suprenum element) and NIL (the infinum element), respectively. The generated lattice is displayed in Fig. 2. On the basis of this lattice, we say that $ec_1 < ec_2$ if $type_{c_1} < type_{c_2}$. Also, it is possible to exhibit the least common generalization lcg, and the greatest common specialization gcs of two element category types ec_1 and ec_2:

$$lcg(ec_1, ec_2) = \min\{type_c \mid type_{c_1} <= type_c \wedge type_{c_2} <= type_c\}; \quad (1)$$

$$gcs(ec_1, ec_2) = \max\{type_c \mid type_c <= type_{c_1} \wedge type_c <= type_{c_2}\}. \quad (2)$$

Analogously, in an instance-level ECG diagram graph each $ei \in EI$ element instance is given as $ei = type_i : caption$, where $type_i \in T'$ and $caption$ is a string representing the name of the corresponding element instance. The members of T' are constructed from the members of T augmented with a number. Thus, an element instance type has the form $type_i = type_c\_n$, where $type_c$ is the corresponding element category type and n is a numeric code, so that $type_i = type_c\_n$ is a unique identifier within the problem domain. The element category type part of an element instance type is denoted by $[type_i] = type_c$, while the numeric code of an element instance type can be obtained as $\{type_i\} = n$.

Definition. *An ECG diagram graph can be defined as $\Gamma = \langle V, A, R\rangle$. V is the set of vertices containing e_i element instances where $[type_i] \in T_{cc} \cup T_{pc}$. A is the set of arrows (directed edges) containing e_i element instances where $[type_i] \in T_{rr} \cup T_{sr}$. R is the set of semantic roles with which the arrows in A are labeled. Thus, the f incidence function assigns an ordered pair of vertices in V and a semantic role in R to each arrow in A, that is $f(a_i) = (v_i, v_j, r_k)$.*

Two element instances ei_1 and ei_2 are said to be equivalent if $type_{i_1} = type_{i_2}$. ECG element instance types can be merged in a domain-specific lattice $(T', \prec)$, whose partial ordering relation $\prec$ can be interpreted as a categorical generalization relation. The top and the bottom elements of the $(T', \prec)$ lattice are UNIV (the suprenum element) and NIL (the infinum element), respectively. On the basis of this lattice we say that $ei_1 \prec ei_2$ if $type_{i_1} \prec type_{i_2}$. Also, it is possible to exhibit the least common generalization lcg, and the greatest common specialization gcs of two differing element instances ei_1 and ei_2 ($ei_1 \neq ei_2$):

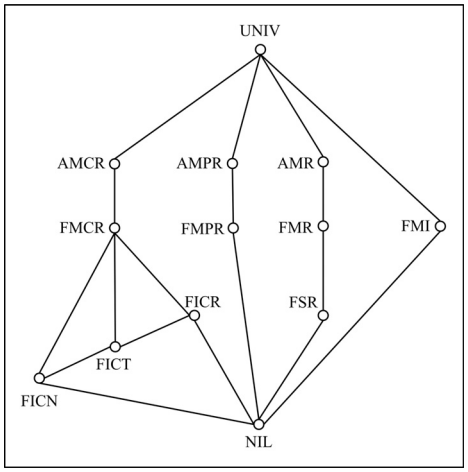


Fig. 2. ECG element category type lattice

$$lcg(ei_1, ei_2) = \min\{type_i \mid type_{i_1} \prec= type_i \wedge type_{i_2} \prec= type_i\}; \quad (3)$$

$$gcs(ei_1, ei_2) = \max\{type_i \mid type_i \prec= type_{i_1} \wedge type_i \prec= type_{i_2}\}. \quad (4)$$

B. Graph Matching and Association

It is possible to insert ECG diagram graphs gradually into an initially empty knowledge base, which is itself another (accumulated) ECG diagram graph. This corresponds to association in the process of conceptualization, that is incorporating new information items into the existing knowledge base. In this operation, the ECG diagram graph to be inserted (Γ_2) must be matched to the knowledge base (Γ_1) according to the following algorithm.

1) The M mapping (alignment) of the two ECG diagram graphs must be performed resulting in μ.
2) If $\mu(\Gamma_1, \Gamma_2) = 1$ then $\Gamma_1 \equiv \Gamma_2$, that is $V_1 = V_2 \wedge A_1 = A_2$.
3) If $\mu(\Gamma_1, \Gamma_2) = 0$ then $V_1 \cap V_2 = \emptyset$. In this case Γ_2 is inserted into the knowledge base in a disjunctive way.
4) If $0 < \mu(\Gamma_1, \Gamma_2) < 1$ then $V_1 \cap V_2 \neq \emptyset$. In this case, if Γ_2 is not a subgraph of Γ_1 then $\forall v_{2i}, a_{2i} \in \Gamma_2 \mid v_{2i}, a_{2i} \notin \Gamma_1$ are inserted into the knowledge base in a conjunctive way.

By definition, the matching operation determines an alignment (i.e. a set of mapping elements M) for a pair of ontologies. A mapping element $m \in M$ is defined as a triplet $\langle ei_1, ei_2, \varphi\rangle$, where

- $ei_1 \in \Gamma_1$ and $ei_2 \in \Gamma_2$ are the aligned element instances of the two ontologies, and

- φ is the correlation between ei_1 and ei_2.

In order to obtain an unambigous (bijective) mapping only the mapping elements where $\varphi \in \{=\}$ are included in the alignment. For describing the result of the matching, an alignment measure is introduced. Let l denote the number of mapping elements in an alignment M. For a given pair of aligned ECG diagram graphs Γ_1 and Γ_2, the fitness value μ is calculated as:

$$\mu(\Gamma_1, \Gamma_2) = \frac{l}{\frac{j+k}{2}}, \tag{5}$$

where j is the number of element instances in graph Γ_1, while k is the number of element instances in graph Γ_2. In this way, the fitness value falls into the $[0, 1]$ interval.

An ECG diagram graph Γ_2 is a subgraph of Γ_1 ($\Gamma_2 \subseteq \Gamma_1$):

- if Γ_1 contains a subgraph Γ_1' which is identical to Γ_2, i.e. $\Gamma_1' \equiv \Gamma_2$, that is $V_1' = V_2 \ \wedge \ A_1' = A_2$; or
- if Γ_1 contains a subgraph Γ_1' which is isomorphic to Γ_2 ($\Gamma_1' \simeq \Gamma_2$).

Definition. *Two ECG graphs are said to be isomorphic ($\simeq$), if one can be obtained from the other by restricting some of its element instances based on the element instance type lattice.*

C. Generalization

The association operation does not involve generalization. It covers only the accumulation of incoming information. However, by the increase of the amount of incoming data the knowledge base would be subtle and computationally intractable without the use of generalization. The generalization algorithm (Fig. 3) gives as result the least common generalized graph that can be obtained from two ECG graphs. This can be achieved formally by finding frequent knowledge patterns (ECG subgraphs) the context of which is similar.

Definition. *Two ECG diagram subgraphs $\gamma_1 \in \Gamma_1$ and $\gamma_2 \in \Gamma_2$ are said to be similar subgraphs if*

- *$\gamma_1 \equiv \gamma_2$ or $\gamma_1 \simeq \gamma_2$, and*
- *they are connected to differing but semantically comparable ECG concept nodes.*

Two similar subgraphs are considered as maximal similar subgraphs if they cannot be extended further without violating the criterion of similarity.

Thus, the maximal similar subgraphs are searched for in Γ_1 and Γ_2. For this, the operation of ECG graph intersection must be introduced. The intersection of two ECG graphs Γ_1 and Γ_2 is the set of identical or isomorphic connected subgraphs. Formally,

$$\begin{aligned} &\Gamma_1 \cap \Gamma_2 = \{\gamma_1, \gamma_2, \ldots, \gamma_k\} \\ &\text{where } \forall \gamma_i \ : \ \gamma_i \in \Gamma_1 \ \wedge \ \gamma_i \in \Gamma_2 \text{ or} \\ &\qquad \gamma_i \in \Gamma_1 \ \wedge \ \gamma_i' \in \Gamma_2 \text{ where } \gamma_i \simeq \gamma_i'. \end{aligned} \tag{6}$$

The extension of the ECG graph intersection operation results in the pairs of similar subgraphs of Γ_1 and Γ_2. Formally,

$$\begin{aligned} &\Gamma_1 \cap^* \Gamma_2 = \{(\gamma_{11}^*, \gamma_{12}^*), (\gamma_{21}^*, \gamma_{22}^*), \ldots, (\gamma_{k1}^*, \gamma_{k2}^*)\} \\ &\quad \text{where } \ \forall \ (\gamma_{i1}^*, \gamma_{i2}^*) \ : \ \gamma_{i1}^* \cup \gamma_{i2}^* = \gamma_i \cup \{ei_1, ei_2\} \mid \\ &\quad \gamma_{i1}^*, ei_1 \in \Gamma_1 \ \wedge \ \gamma_{i2}^*, ei_2 \in \Gamma_2 \ \wedge \ \gamma_i \in \Gamma_1 \cap \Gamma_2. \end{aligned} \tag{7}$$

Maximal similar subgraphs are then obtained from merging all similar subgraphs with the same root nodes, that is similar subgraphs having the same pair of differing concepts.

$$\begin{aligned} &\{\max(\gamma_{i1}^*, \gamma_{i2}^*)\} = \{\cup(\gamma_{i1}^*, \gamma_{i2}^*)\} \mid \\ &\quad \forall(\gamma_{i1}^* \cup \gamma_{i2}^*) = \gamma_{i_j} \cup \{ei_1, ei_2\}. \end{aligned} \tag{8}$$

Require: Γ_1, Γ_2
  μ = Match(Γ_2, Γ_1)
  if $\mu = 1$ **then**
    Return
  end if
  if $\mu = 0$ **then**
    Insert Γ_2 into Γ_1
  end if
  if $0 < \mu < 1$ **then**
    if $\Gamma_2 \subseteq \Gamma_1$ **then**
      Return
    else
      Search for maximal similar subgraphs in Γ_1, Γ_2
      for all (γ_1^*, γ_2^*) **do**
        if $ei_1 >< ei_2$ **then**
          if $lcg(ei_1, ei_2) \neq UNIV$ **then**
            if $lcg(ei_1, ei_2) \notin \Gamma_1$ **then**
              Insert $lcg(ei_1, ei_2)$ into Γ_1
            end if
            if $lcg(ei_1, ei_2) \neq ei_1$ **then**
              Connect ei_1 to $lcg(ei_1, ei_2)$ by FMI
            end if
            Update relations of ei_1 in Γ_1
            if $ei_2 \notin \Gamma_1$ **then**
              Insert ei_2 into Γ_1
            end if
            if $lcg(ei_1, ei_2) \neq ei_2$ **then**
              Connect ei_2 to $lcg(ei_1, ei_2)$ by FMI
            end if
          end if
        end if
      end for
      for all $ei \in \Gamma_2$ **do**
        if $ei \notin \Gamma_1$ **then**
          Insert ei into Γ_1
        end if
      end for
      Update relations of ei_2 in Γ_1
    end if
  end if
  return Γ_1

Fig. 3. The generalization algorithm

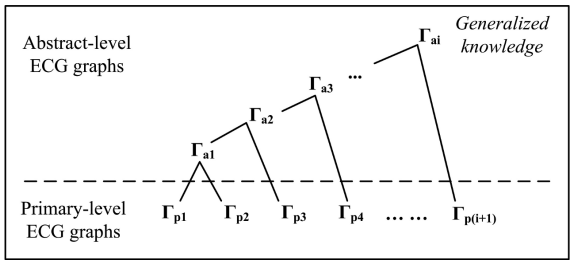


Fig. 4. The process of generalization

For the differing concepts semantic comparability should be checked. Two element instances are said to be semantically comparable if $lcg([type_{i_1}], [type_{i_2}]) \neq UNIV$ on the basis of the element category type lattice. If this is the case, instead of the differing concepts a new concept is introduced from the element instance type lattice determined as $lcg(ei_1, ei_2)$, if it is not the $UNIV$ top element. It is possible, that $lcg(ei_1, ei_2)$ results in one of its arguments. In this case actually no insertion occurs. The differing concepts are connected to the new concept via specialization relationships and the other relationships originally in connection with the differing concepts should also be updated.

At the end of the generalization process (Fig. 4) stands the generalized knowledge an agent can obtain from the samples observed. This can be formulated as recursively determining the least common generalization of the previous abstract-level ECG graph and a primary-level ECG graph, that is

$$\Gamma_{a_i} = \text{lcg}(\Gamma_{a_{i-1}}, \Gamma_{p_{i+1}}). \tag{9}$$

D. Test Environment

Let us assume that the environment of the learning agent is a microworld of 2D shapes and each observation includes two objects in a binary relation. The relevant segment of the element instance type lattice generated from the microworld is shown in Fig. 5.

For practical purposes, two more generalization rules have been introduced. Firstly, abstract concepts can not be derived directly from instance concepts, and secondly instance concepts can be omitted from the generalized graph if they are completely substituted by a class concept. The demonstration of the process of conceptualization (association and generalization) can be seen in Fig. 7.

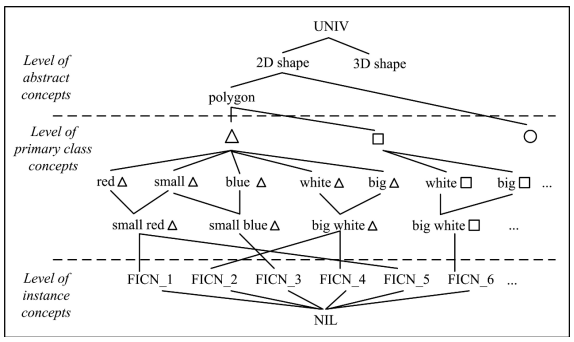


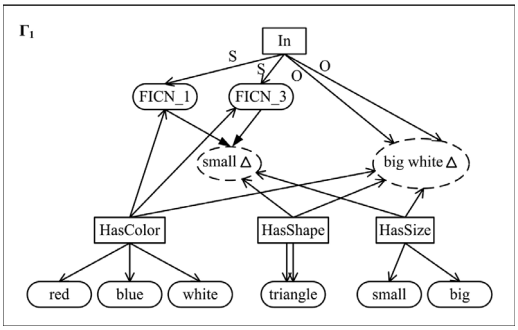
Fig. 5. A segment of the element instance type lattice

V. Specialization

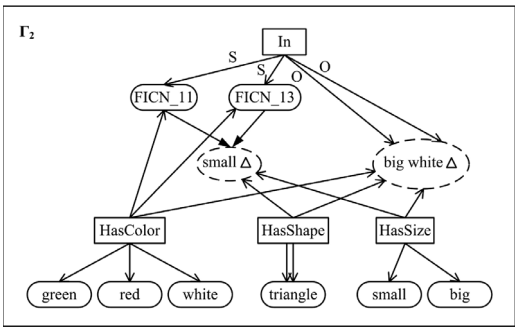
Similarly to the formulation of the least common generalization of two ECG graphs, it is possible to give the greatest common specialization of two ECG graphs. This latter can be defined as the maximal common restriction of the two graphs, that is the union of the maximal similar subgraphs of the two graphs. Formally,

$$gcs(\Gamma_1, \Gamma_2) = \bigcup(\{\max(\gamma^*_{i1}, \gamma^*_{i2})\}). \tag{10}$$

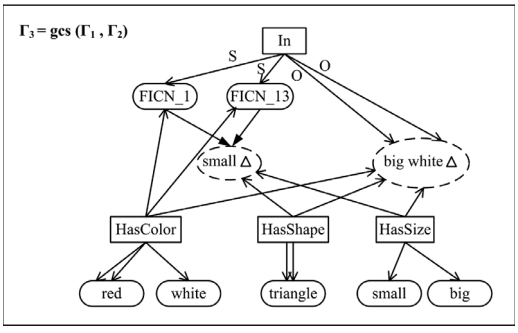
For the demonstration of specialization, see the next example in Fig. 6. Given two ECG diagram graphs Γ_1 and Γ_2, their greatest common specialization is shown by Γ_3.



(a)



(b)



(c)

Fig. 6. Demonstration of specialization

VI. Conclusion

The paper has shown how the ECG graph-based knowledge base of the grammar learning agent examined is built up from primary-level ECG graphs. The operation of association – that is matching and connecting ECG diagram graphs – can always be accomplished, but generalization – that is the introduction of higher-level concepts – does not necessarily occur in each step. Also, it is possible that the greatest common specialization of two ECG graphs results in an empty graph.

Nevertheless, the least common generalization and the greatest common specialization of two ECG graphs always exist and can be computed. Therefore, the definition of the $\prec$ relation on element instances can be extended to a partial relation $\preceq$ on ECG diagram graphs, according to which $\Gamma_1 \preceq \Gamma_2$ if graph Γ_1 is more 'specialized' than graph Γ_2.

Let $\mathbf{\Gamma}$ denote the set of primary-level ECG diagram graphs (representing environment snapshots) that are matched to and incorporated in the knowledge base of the agent, and $\mathbf{\Gamma}(\mathbf{A})$ denote the set of accumulated ECG diagram graphs resulting from the conceptualization (association and generalization) steps executed.

Conclusion. *The $\preceq$ relation effects a lattice structure on the union of $\mathbf{\Gamma}$ and $\mathbf{\Gamma}(\mathbf{A})$.*

The top element of the lattice $(\mathbf{\Gamma} \cup \mathbf{\Gamma}(\mathbf{A}), \preceq)$ symbolizes the accumulated knowledge of the agent at the end of the conceptualization process. The bottom element is NIL, the infinum element.

Acknowledgment

This research was carried out as part of the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund.

References

[1] E. Baksa-Varga and L. Kovács, "Knowledge base representation in a grammar induction system with Extended Conceptual Graph," *Transactions on Automatic Control and Computer Science, Scientific Bulletin of "Politehnica" University of Timisoara, Romania*, vol. 53, no. 67, pp. 107–114, 2008.

[2] J. Fargues, M. Landau, A. Dugourd, and L. Catach, "Conceptual graphs for semantics and knowledge processing," *IBM J. Res. Develop.*, vol. 30, no. 1, January 1986.

[3] E. Baksa-Varga and Kovács, "Semantic representation of natural language with Extended Conceptual Graph," *Journal of Production Systems and Information Engineering*, vol. 5, pp. 19–39, 2009.

[4] L. Kovács and E. Baksa-Varga, "Logical representation and assessment of semantic models for knowledge base representation in a grammar induction system," *Journal of Computer Science and Control Systems, University of Oradea, Romania*, pp. 48–53, 2008.

[5] J. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley, 1984.

[6] M. Ponsen, M. Taylor, and K. Tuyls, "Abstraction and generalization in reinforcement learning: A summary and framework," *ALA Workshop, Adaptive and Learning Agents (LNAI Journal)*, 2010.

[7] L. Kovács, "Concept lattice structure with attribute lattices," *Production Systems and Information Engineering*, vol. 4, pp. 65–81, 2006.

[8] B. Ganter and R. Wille, *Formal Concept Analysis, Mathematical Foundations*. Springer Verlag, 1999.

[9] R. Godin, R. Missaoui, and H. Alaoui, "Incremental concept formation algorithms based on Galois lattices," *Computational Intelligence*, vol. 11, no. 2, pp. 246–267, 1995.

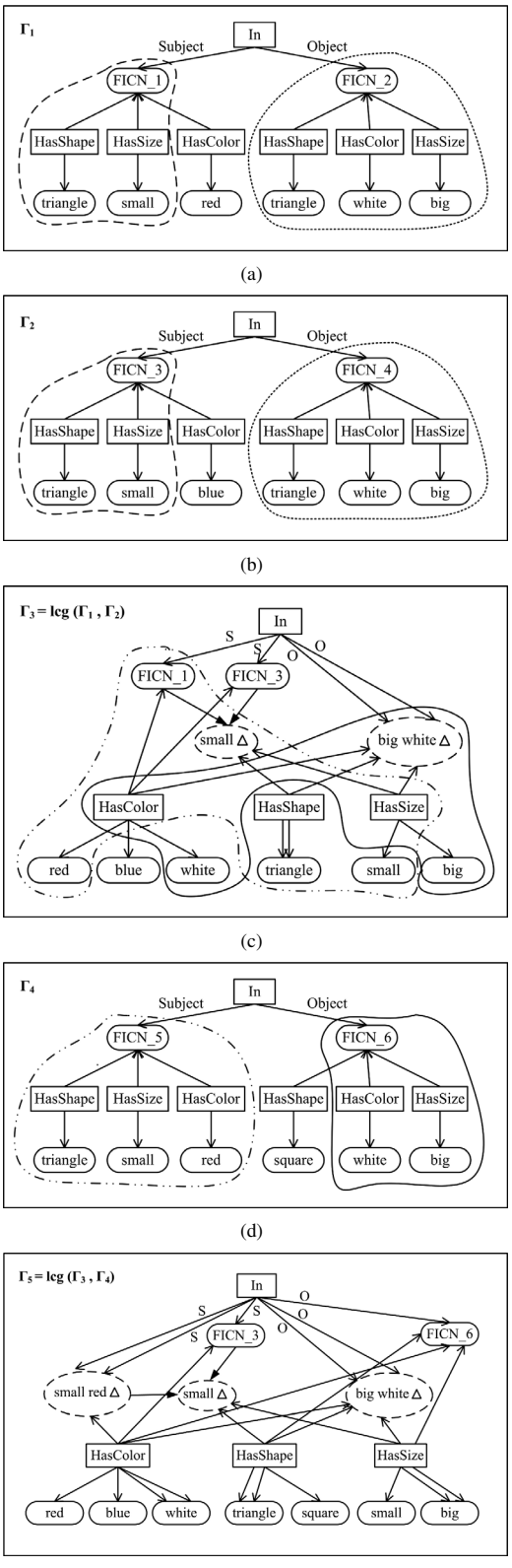


Fig. 7. Demonstration of conceptualization

Multi-level Sequence Mining Based on GSP

Michal Šebek, Martin Hlosta, Jan Kupčík, Jaroslav Zendulka, Tomáš Hruška

Department of information systems
Faculty of Information Technology
Brno University of Technology Technology
Brno, Czech Republic
Email: {isebek,ihlosta,ikupcik,zendulka,hruska}@fit.vutbr.cz

***Abstract*—Mining sequential patterns is an important problem in the field of data mining and many algorithms and optimization techniques have been published to deal with that problem. An GSP algorithm, which is one of them, can be used for mining sequential patterns with some additional constraints, like gaps between items.**

Taxonomies can exist upon the items in sequences. It can be applyed to mine sequential patterns with items on several hierarchical levels of the taxonomy. If a more general item appears in apattern, the pattern has higher or at least the same support as the one containing the corresponding specific item. This allows us to mine more patterns with the same minimal support parameter and to reveal new potentially useful patterns.. This paper presents a method for mining multi-level sequential patterns. The method is based on the GSP algorithm and generalization of more specific sequences based on the information theory.

***Index Terms*—Sequence pattern mining, GSP, taxonomy.**

I. Introduction

A great amount of data is being collected and stored in databases for various purposes. After years the size of these databases became enormous. In such amount of data hidden and interesting patterns may occur. Data mining, also known as knowledge discovery in databases, is a field of study to find these patterns in data. One of the typical data mining task is discovery of frequent patterns and *association analysis*. The latter one is seeking for rules in the form $antecedent \rightarrow consequent$, which occur in the data often enough and are so called strong. The strength means that the conditional probability $P(consequent|antecedent)$ is above a given threshold. The association analysis is abundantly used in market basket analysis to discover habits of customers. Its results can be employed for product recommendation. An example of such a rule can be $TV \rightarrow DVD\,player$ telling that when customers buys TV it's likely that a DVD player will appear in their market basket too.

If time information occurs in the data, not only association analysis but also a *sequential patterns mining* task can be sspecified. In such a case, the task is to reveal frequently occurring sequences in a sequence database. According to previously mentioned market basket analysis, an example of such a sequence pattern is $\langle TV\ DVD\_player \rangle$. This means that customers very often buy a TV set and later a DVD player.

In addition, a one or multiple taxonomies of items can be stored in the database. This allows mining patterns with items on different levels of hierarchy specified by the taxonomies. Then, the resulting pattern set can contain items from more general levels that might not be directly stored in the database. This can provide the analyst better insight of the data and reveal patterns that he wouldn't get with a pure sequence mining algorithm. It also allows us to set the minimum support parameter to a higher value and to get results containing more database sequences, because sequences with more general items have at least the same support as their more specific variants. Some algorithms can only find patterns, where all items in one pattern are on the same level of the hierarchy. They are referred to as *intra-level patterns*. If items of the pattern can be on different levels of hierarchy, they are called *inter-level* or *level-crossing patterns*. Our method is capable of revealing both intra-level and inter-level sequence patterns.

The remainder of the paper is organized as follows. In section II there is formally defined the problem of mining sequential patterns with taxonomies and terms related to that problem and to the information theory. Algorithms related to our work are described in section III. The proposed method for mining sequential patterns with taxonomies is described in section IV. In section V we present performance evaluation of our method.

II. Problem definition

In this section we present definitions of notions for mining sequential patterns with taxonomies from databases and the problem is formally defined. The section begins with terms related to sequential pattern mining, then focuses on taxonomy and ends with terms of the information theory necessary for our algorithm.

A. Sequential pattern mining

Definition 1 (Itemset)**.** Let $I = \{I_1, I_2, I_3 \dots I_k\}$ be a set of all items, that are stored in the database. Then *itemset* $i = (x_1 x_2 \dots x_m)$ is a nonempty subset of I containing m distinct items. Thus itemset is an unordered list of items, but with no loss of generality we can assume that items in the itemset are ordered lexicographically. Given $I = \{a, b, c, d, e\}$ an example of an itemset is (abd).

Definition 2 (Sequence)**.** A *sequence* $s = \langle e_1 e_2 e_3 \dots e_n \rangle$ is an ordered list of n itemsets, sometimes called as elements or events. Based on this notation, example of such a sequence

is $s_1 = \langle av(ab)(ce)d(edgi)\rangle$. When the itemset contains only one item, the braces can be omitted as it is shown in this example. The *length* of a sequence is considered as a number of instances of its items. The sequence of length l is called *l-sequence*. The length of previously mentioned sequence is 11, thus it is called *11-sequence*. A sequence $\alpha = \langle a_1 a_2 \dots a_n\rangle$ is a *subsequence* of sequence $\beta = \langle b_1 b_2 \dots b_m\rangle$ if there exist integers $1 \leq j_1 < j_2 < \dots < j_n \leq m$ such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \dots, a_n \subseteq b_{j_n}$. We denote it $\alpha \sqsubseteq \beta$ and β is a supersequence of α [5].

Definition 3 (Sequence DB)**.** A sequence database D is a set of tuples $\langle SID, s\rangle$, where *SID* is the identification of a sequence and s is the sequence. The support of the sequence s_1 is defined as a number of tuples in the database in which sequences are supersequences of s_1. Formally, the support of sequence s_1 is

$$support(s_1) = |\{\langle SID, s\rangle | (\langle SID, s\rangle \in D) \wedge (s_1 \sqsubseteq s)\}|. \quad (1)$$

Definition 4. Sequence pattern is defined as a frequent sequence, support of which is greater than the *minimum support* parameter, which is provided by the user.

Based on the defined terms, we can formally define the problem of mining sequential patterns as follows: Given a sequential database $D = \{i_1, i_2 \dots i_n\}$, where each i_i in this database is an itemset, and the minimal support parameter *min_sup,* we are looking for all the sequences with support $\geq min\_sup$.

B. Taxonomy

Definition 5 (Taxonomy)**.** A *taxonomy structure* is an ordered directed tree. The taxonomy structure has $l+1$ levels. The node on the level $h = 0$ is called *root node*, the nodes on a level h where $0 \leq h \leq l$ are *internal nodes* and the nodes on the level $h = l+1$ are *leaf nodes*. The edges between nodes form *is-a relation*, the *specialization* of terms related to nodes is from the root to the leaf nodes. The *generalization* is from the leaf node to the root. An example of a taxonomy structure is depicted in figure 1, which represents some food products. The cheese *is a* dairy product and all dairy products and pastry are food. In our problem the items can be considered as nodes in the taxonomy structure.

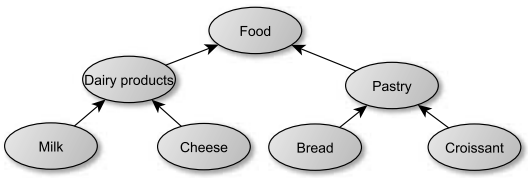


Figure 1. Example of a food taxonomy with three levels and root containing all food.

Definition 6 (Parents)**.** Given an item i on a level h in the taxonomy structure, the *parent* of i, denoted as $parent(i)$, is its generalized item on level $h-1$, the $ancestor(i)$ is a set of all generalized items of i on a level l where $0 \leq l < h$.

Given an element $e = \{i_1, i_2, \dots, i_n\}$, a set of *parent elements* of the element e is a set of the elements which are same as e but exactly one of the items is generalized. This is defined as

$$\begin{aligned} parent(e) \quad = \quad & \{\{j_1, j_2, \dots, j_n\} | \exists k : i_k \in e \\ & \wedge parent(i_k) = j_k \\ & \wedge \forall l \neq k : i_l = j_l\}. \end{aligned} \quad (2)$$

Notice that we took advantage of the property that items in elements are lexicographically ordered. Based on the definition of parent of an element, we can define the parent of a sequence. Given a sequence $s = \langle e_1 e_2 \dots e_n\rangle$, the parent of s is the set of sequences which are the same as the sequence s but one of their element is replaced by its parent. This can be defined as

$$\begin{aligned} parent(s) \quad = \quad & \{\langle f_1 f_2 \dots f_n\rangle | \exists k : e_k \in s \\ & \wedge parent(e_k) = f_k \\ & \wedge \forall l \neq k : e_l = f_l\}. \end{aligned} \quad (3)$$

Based on the definition of the parent of a sequence, the ancestor of the sequence s is defined as the set $ancestor(s)$ as follows:

$$\begin{aligned} & 1. \quad parent(s) \in ancestor(s) \\ & 2. \quad \forall x \in ancestor(s) : parent(x) \in ancestor(s). \end{aligned} \quad (4)$$

Notice that the sequence s and all the sequences in the $ancestor(s)$ has one common ancestor. This sequence consists of elements with root items of the items in sequence s. We denote this sequence as *root sequence*.

Example 1. These notions can be well understood by looking at the figure 2 with two taxonomy structures for two root items A and B. Item A is an ancestor and a parent of items a and a'. The bottom of the figure depicts ancestors for the sequence $\langle a(a'b)\rangle$. The arrows in the figure represent which nodes are parents to the selected node. Sequences $\langle a(a'B)\rangle, \langle A(a'b)\rangle, \langle a(Ab)\rangle$are parents of the sequence $\langle a(a'b)\rangle$, because one of their elements is the parent of an element in $\langle a(a'b)\rangle$, for example $(a'B)$ is the parent of the $(a'b)$. All of the sequences in nodes above sequence$\langle a(a'b)\rangle$ are its ancestors and the sequence $\langle A(AB)\rangle$ is its root sequence.

Definition 7. The *generalized support gen_supp* is based on the definition of support in definition 2. It's only necessary to redefine it, if an element e_1 is a subset of an element e_2. For this purpose we define the generalized subset relation $\subseteq_g$ as

$$\begin{aligned} e_1 \subseteq_g e_2 \quad \Leftrightarrow \quad & \forall i \in e_1 : i \in e_2 \vee \\ & \exists j \in e_2 : i \in ancestor(j). \end{aligned} \quad (5)$$

A sequence $\alpha = \langle a_1 a_2 \dots a_n\rangle$ is a *generalized subsequence* of sequence $\beta = \langle b_1 b_2 \dots b_m\rangle$ if there exist integers $1 \leq j_1 < j_2 < \dots < j_n \leq m$ such that $a_1 \subseteq_g b_{j_1}, a_2 \subseteq_g b_{j_2}, \dots, a_n \subseteq_g$

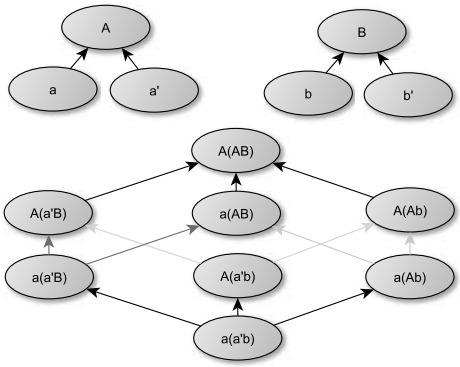


Figure 2. Hierarchy in sequences.

SID	sequence
1	$\langle ab(a'b)b\rangle$
2	$\langle b(aa')b(ab')aa\rangle$
3	$\langle (ab)ba'a(bb')\rangle$
4	$\langle b'aa(a'b')b\rangle$
5	$\langle (ab)(ab)(ab)\rangle$

Table I
EXAMPLE OF SEQUENCE DATABASE D WITH 5 SEQUENCES, $|D| = 5$.

b_{j_n}. We denote $\alpha \sqsubseteq_g \beta$. For completeness, the definition of *the generalized support* of a sequence s_1 is

$$gen\_supp(s_1) = |\{\langle SID, s\rangle | (\langle SID, s\rangle \in D) \wedge (s_1 \sqsubseteq_g s)\}|. \quad (6)$$

Example 2. An example of how the generalized support of a sequence is computed can be shown on a sequence database from table I and taxonomy structures over items a, a', b, b' from figure 2. For sequence $s_1 = \langle a(Ab)\rangle$ generalized support $gen\_supp(s_1) = 2$, because it is generalized sequence of tuples with $SID \in \{1, 5\}$. If we take a parent of s_1, the sequence $s_2 = \langle a(AB)\rangle$, the generalized support will be even higher. In addition to s_1, s_2 is contained in the tuple with $SID = 4$, because $\langle a(AB)\rangle \sqsubseteq_g \langle b'aa(a'b')b\rangle$. The higher support is the consequence of $(AB) \subseteq_g (a'b')$. Element $(Ab) \not\subseteq_g (a'b')$, so that is why sequence s_1 has lower support than s_2.

Having the definition of the taxonomy structure, we can define the task of mining sequential patterns with taxonomy in almost the same way as the problem without taxonomies. In addition, the presence of the taxonomies over the items in a sequence database causes that more general terms will occur in the result.

However not all of these patterns are interesting for an analyst. The question is: which of the generalized sequences are interesting to incorporate them into result set of patterns? This decision can be based on information theory which is described in more details in section IV.

III. RELATED WORK

The first algorithm for mining sequential patterns *AprioriAll* was published by Agrawal and Strikant in 1995 in [2]. It was the modification of the well-known *Apriori* algorithm for mining association rules such that it can mine sequential patterns. The general idea of both *Apriori* and *AprioriAll* is based on a candidates generate-and-test framework. The same authors then presented algorithm *Generalized Sequence Patterns (GSP)* in [12]. Our proposed algorithm is based on *GSP*, so this algorithm will be described in more detail.

A. GSP algorithm

As mentioned, the general idea of this algorithm is based on the Apriori property and candidates generation and testing approach. The apriori property states: *Every nonempty subsequence of a sequential pattern is a sequential pattern [5].*

The algorithm works in several phases. In each of them it makes a pass over the sequence database:

1) In the first phase, the support of items is counted and those having it higher than the *min_sup* are inserted in the resulting set L_1 containing *frequent 1-sequences*.
2) Each of the next phases iconsists of two sub-steps.
   a) At first, in the *join step*, the *candidate set* C_k is generated from the set of frequent items from the previous phase L_{k-1}. The candidate sequences C_k contain one more item than sequences in L_{k-1}. Candidates of length two are generated from items in the manner, that either they occur in one transaction, thus forming an element, or one is after the other, thus forming a sequence. For example, for items a, b either $\langle (ab)\rangle$ or $\langle ab\rangle$can be generated . Candidates of higher length are generated from s_1 and s_2 such that if the first item of s_1 and the last item of s_2 are omitted, the resultant subsequences are the same. The sequences s_1 and s_2 are called *contiguous* and those *k-sequences* which has non-frequent contiguous subsequences are pruned.
   b) After that, in *counting step*, the support of each candidate is counted resulting in frequent set L_k. The algorithm terminates in phase n when no candidate sequence satisfies the minimum support condition or no candidate sequence can be generated because of pruning step. The result set is composed of $\bigcup_{k=1}^{n-1} L_k$.

In comparison to *AprioriAll*, the GSP incorporates time constraints, sliding time windows and taxonomies in sequence patterns and thus allowing to mine for *generalized* patterns. The patterns with given time constraints and sliding time windows are achieved by a procedure which detects if the candidate is a subsequence of any sequence in database satisfying the constraints.

B. Other sequence pattern mining algorithms

Some modifications of algorithm GSP were published later, for example *PSP* in [9] introduces a different tree structure for

maintaining candidates. In the category of algorithms based on candidate generating and testing, it's worth mentioning algorithm *SPADE [14]* which uses for mining vertical representation of sequence database; and also *SPAM* [3], which is similar to SPADE but uses internal bitmap structure for database representation.

The next family of algorithms is based on the *pattern-growth* principle. The key idea of these algorithms is that at first, it is created a representation of the sequence database to partition the space and then the search space is often traversed in depth-first manner to generate less candidate sequences. The most famous representative is the *PrefixSpan* published by Pei et al. in [10], which uses projected database with respect to some prefix for database representation.

The most recent algorithms are based on *early pruning*. The algorithms try to prune the searched space in the earliest phases based on the *position induction.* They store the last positions of items in database sequences and use this information to prune candidates which can't be appended to the current prefix. The rest of the algorithm is the same as in pattern-growth based. *LAPIN* [13] is the typical representative of these algorithms.

C. Hierarchies in sequence pattern mining

The authors of GSP presented the way how to incorporate the taxonomies into the process of sequential pattern mining [12]. The idea is based on replacing all the sequences in database with *"extended-sequences"*. In this extended form, in addition to the item, the information of all its ancestors is stored. For example sequence $\langle$(milk,bread)(croissant)$\rangle$ from the taxonomy in figure 1 is replaced by $\langle$(milk, pastry, dairy product, food) (croissant, pasrty, food)$\rangle$. Then the GSP algorithm is the same as was mentioned before on these *"extended-sequences"*. Although the authors proposed two optimizations, this approach requires much more space for storing the database sequences. It also allows to mine all the frequent sequences, even those which could be uninteresting.

In [11], the authors were using their framework for multidimensional sequence mining. They presented the *HYPE* algorithm to incorporate hierarchies into this framework and mine for multidimensional sequences over several levels of hierarchy.

In [6] T. Huang presented the concept of *fuzzy* multi-level sequential patterns. In this concept the item is allowed to belong among more general concepts, for example tomato can be considered either as fruit or vegetable. This relationship can be represented by a value between 0 and 1. They proposed the algorithm based on the divide-and-conquer strategy and an efficient algorithm to mine fuzzy cross-level patterns.

IV. The hGSP algorithm

In this section we describe our algorithm *hGSP (hierarchical-GSP)* for mining level crossing sequence patterns. The algorithm is based on GSP algorithm [12]. The objective of the algorithm is to get the complete set of maximally concrete frequent sequences. The concreteness measure will be evaluated using information theory explained in following subsection. In following text we use *support* term in meaning of our defined *generalized support*.

A. Algorithm concept

The main idea of our algorithm is that if a sequence s has support $gen\_supp(s)$, there can exist a generalized sequence $s_g \in parent(s)$ such that $gen\_supp(s_g) > gen\_supp(s)$. This can be applied repeatedly. Notice that $\forall s_g \in parent(s) : gen\_supp(s) \leq gen\_supp(s_g)$. Unfortunately, during generalization some information is being lost. Because of this, the quality of mined generalized frequent items strictly depends on the selection of a the generalized sequence from the set of generalized sequences. The concepts of information theory is used for this purpose. Generally, we expect that more specific sequence s is more important result than it's generalized form s_g because the generalized s_g is more expectable in the result set. This corresponds with meaning of information content.

Definition 8. The Shanon *information content* [8] of value x with probability $p(x)$ is defined as

$$h(x) = \log_2 \frac{1}{p(x)}. \tag{7}$$

The probability $p(s)$ that sequence s occurs in source database D is

$$p(s) = \frac{gen\_supp(s)}{|D|}. \tag{8}$$

The *information content of sequence* s in database D is

$$h(s) = \log_2 \frac{1}{\frac{gen\_supp(s)}{|D|}} = -\log_2 \frac{gen\_supp(s)}{|D|}. \tag{9}$$

For a sequence s, the dependence between information content $h(s)$ and generalized support $gen\_supp(s)$ causes that if the generalization from s to s_g is performed and $gen\_supp(s_g) > gen\_supp(s)$, then $h(s_g) < h(s)$. Therefore the generalization should be performed only if the candidate sequence is not frequent (i.e. $gen\_supp(s) < min\_supp$) or the GSP algorithm cannot perform *join* of two candidate sequences with joinable ancestors.

Definition 9. *Concreteness* The sequence s is more *concrete* than another sequence s_1 if $(h(s_1) < h(s)) \wedge (ancestor(s) \cup s) \cap (ancestor(s_1) \cup s_1) \neq \emptyset$.

B. Algorithm hGSP

The hGSP algorithm uses the modified *join step* and *pruning step* of the GSP algorithm. The rest of algorithm remains the same.

The *join step* is modified for generating candidates of length $k = 3$ and more. Let's have a pair of frequent sequences s_1 and s_2 of length $k - 1$. A join can be performed if subsequences of s_1 after omitting the first item and s_2 after omitting the last onehave a common *ancestor* sequence. Then the joined

sequence of length k is composed from first item of s_1, most concrete ancestor sequence of common part and the last item of s_2. The last item is added same as in GSP.

The support of candidates is being counted almost the same as in the original GSP. The only difference is that we use $gen\_supp(s)$ defined in Definition 7 instead of common support. Thus only modified procedure for checking if a candidate is subsequence of a given database sequence is used.

The modification of *pruning step* is shown in Algorithm 2. The algorithm uses method for finding the most concrete generalization set of sequence which is described in Algorithm 1. We have formulated theorem, that is not necessary to evaluates all information contents with logarithm functions but it is possible to only compare ratios of supports of sequences and theirs generalized forms.

Let's have a sequence s and it's generalized form s_1. Information contents of these sequences are $h(s) = -\log_2 \frac{gen\_supp(s)}{|D|}$ and $h(s_1) = -\log_2 \frac{gen\_supp(s_1)}{|D|}$. The information lost during generalization of s to s_1 is $\Delta h = h(s) - h(s_1)$. It follows that

$$\Delta h = \log_2 \left(\frac{\frac{gen\_supp(s_1)}{|D|}}{\frac{gen\_supp(s)}{|D|}} \right) = \log_2 \left(\frac{gen\_supp(s_1)}{gen\_supp(s)} \right). \tag{10}$$

The generalization of s with the smallest information loss is found because then the sequences will be the most concrete. Therefore the algorithm minimizes ratio $\frac{gen\_supp(s_1)}{gen\_supp(s)}$.

Algorithm 1 Method *find_generalization()*

Input: Candidate sequence s
Output: The set of the most concrete generalizations G_s.
Method:

```
G_s = {};
min_supp_ratio = +∞;
foreach (p_s ∈ parent(s)) do:
  ratio = gen_supp(p_s)/gen_supp(s);
  if (gen_supp(p_s) <> gen_supp(s)
        ∧ratio < min_supp_ratio) then
    G_s = {p_s};
    min_supp_ratio = ratio;
  elseif (ratio = min_supp_ratio) then
    G_s = G_s ∪ {p_s};
  endif;
endforeach;
return G_s;
```

V. Experiments

We have used synthetic datasets created by the generator described in [1] with our hierarchical extension. The evaluation was performed on PC Intel Core Duo 2.66 GHz, 2GB RAM, OS Windows XP 32-bit. The implementation of hGSP algorithm was integrated into MS Analysis Services.

Algorithm 2 hGSP Pruning Step

Input: The set of candidates - C_k, minimal support min_supp
Output: The set of frequent sequences L_k of length k
Method:

```
L_k = {};
foreach s_c ∈ C_k do:
  C'_k = {s_c};
  sequence_added = false;
  while (sequence_added = false ∧ |C'_k| > 0) do:
    G_s = {};
    foreach s ∈ C'_k do:
      if gen_supp(s) ≥ min_supp then
        L_k = L_k ∪ {s};
        sequence_added = true;
      else
        G_s = G_s ∪ find_generalization(s);
      endif;
    endforeach;
    C'_k = G_s;
  endwhile;
endforeach;
return L_k;
```

A. Performance evaluation

The performance test is focused on the execution time of the algorithm. We did not compare execution times with GSP because hGSP generates much more candidates. Also, GSP generates shorter sequences and it is finished after sequences of length 2 in most cases. Therefore, we show only execution times of hGSP. Parameters of datasets were:

- *average DB sequence length = 25,*
- *hierarchy count = 25,*
- *average hierarchy depth = 3,*
- *frequent sequences count = 0.2 % of $|D|$*
- *and average length of frequent sequences = 7.*

Results of experiment are shown in Figure 3. The algorithm was executed on datasets of a different size and the execution time was measured. The number of transaction items ranged from $25,000$ in a database of $1,000$ sequences up to $250,000$ transaction items in a database of $10,000$ sequences. The plot shows that processing time increases linearly with the dataset size because the algorithm checks if each candidate sequence is subsequent of any database sequence.

B. Generalization evaluation

The generalization evaluation experiment is focused on generalization property of hGSP algorithm. The methodology of the evaluation is following. It is known that the number of frequent sequences in results strictly depends on the minimal support value. Therefore, we have compared the number of candidates and mined frequent sequences for different values of the minimal support. The results were measured for relative supports from $20\,\%$ up to $80\,\%$. Predefined parameters of frequent sequences in the dataset were: the average length 5

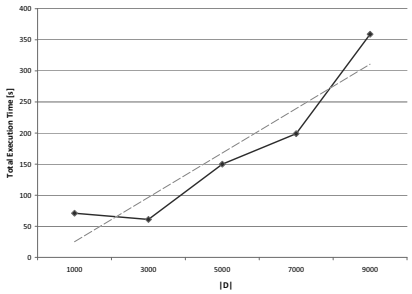


Figure 3. Average processing time of one candidate sequence.

and support $50\,\%$. Results of hGSP are shown in comparison to GSP results.

The results in Figure 4 show that the number of candidate/frequent sequences increases exponentially with descending value of minimal support for both hGSP and GSP. In general, hGSP creates more than 10 times more frequent sequences than the GSP because of the generalization property. This is the main substance for analyst – the hGSP does not prune important candidate sequences if there is a possibility to generalize them. In addition, Figure 5 shows that the hGSP creates sequences with more items than GSP.

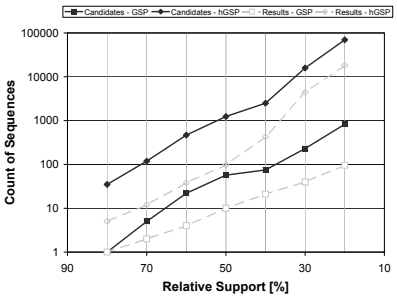


Figure 4. Numbers of candidate and frequent sequences with dependancy on minimal support

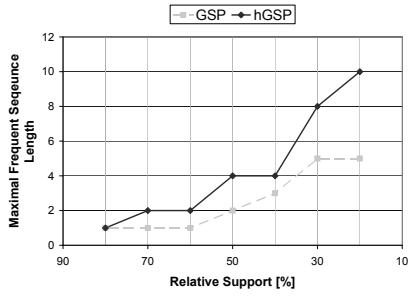


Figure 5. Lengths of frequent sequences with dependancy on minimal support

VI. Conclusions

In this paper we presented a new way how to incorporate taxonomies into the process of mining generalized sequential patterns. Our approach is based on information theory and modifies the well known GSP algorithm. The algorithm is trying to generalize sequences only if it is requried. This is because of either low support of the sequence or inability to join two sequences with joinable ancestors. In the experiments, we showed that in comparison to GSP, the algorithm was able to find more patterns and patterns of higher length thanks to the generalization.

In the future work, we want to focus on optimization of the algorithm. Because the hGSP generates more candidates, it is naturally slower than GSP so we will also try to apply our ideas to more efficient algorithm such as PrefixSpan.

Acknowledgement

This work has been supported by the research funding TAČR TA01010858, BUT FIT grant FIT-S-11-2 and by the Research Plan No. MSM 0021630528.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proc. of the VLDB Conference.*, pages 487–499, Santiago, Chile, 1994. Expanded version available as ABM Research Report RJ8939.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. pages 3–14, 1995.

[3] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 429–435, New York, NY, USA, 2002. ACM.

[4] Jiawei Han and Yongjian Fu. Discovery of multiple-level association rules from large databases. In *In Proc. 1995 Int. Conf. Very Large Data Bases*, pages 420–431, 1995.

[5] Jiawei Han and Micheline Kamber. *Data mining: concepts and techniques*. The Morgan Kaufmann series in data management systems. Elsevier, 2006.

[6] Tony Cheng-Kui Huang. Developing an efficient knowledge discovering model for mining fuzzy multi-level sequential patterns in sequence databases. *Fuzzy Sets Syst.*, 160:3359–3381, December 2009.

[7] Nizar R. Mabroukeh and C. I. Ezeife. A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv.*, 43:3:1–3:41, December 2010.

[8] David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambrifge University Press, 2003.

[9] F. Masseglia, F. Cathala, and P. Poncelet. The psp approach for mining sequential patterns. pages 176–184, 1998.

[10] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. on Knowl. and Data Eng.*, 16:1424–1440, November 2004.

[11] Marc Plantevit, Anne Laurent, and Maguelonne Teisseire. HYPE: mining hierarchical sequential patterns. In *DOLAP 2006, ACM 9th International Workshop on Data Warehousing and OLAP*, pages 19–26, November 2006.

[12] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, 1996. Springer-Verlag.

[13] Zhenglu Yang, Yitong Wang, and Masaru Kitsuregawa. Lapin: effective sequential pattern mining algorithms by last position induction for dense databases. In *Proceedings of the 12th international conference on Database systems for advanced applications*, DASFAA'07, pages 1020–1023, Berlin, Heidelberg, 2007. Springer-Verlag.

[14] Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. In *Machine Learning*, pages 31–60, 2001.

A model for Command and Control Information Systems Semantic Interoperability

Darko Galinec
Department of Informatics and Computing
Zagreb Polytechnic for Technical Sciences
Zagreb, Croatia
darko.galinec@tvz.hr

Viliam Slodičák
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Košice, Slovakia
viliam.slodicak@tuke.sk

Abstract - **In presence of rapid development of information and communications technology (ICT) able to increase data processing, much more data can be stored, used and/or mediated and disseminated. Today, information and knowledge become a resource of strategic importance in complex systems, both business and military. On the other hand the question of credibility of the collected data arises: complex systems have to deal with an increasing access to information with less knowledge about their origins and their quality. Due to this lack of information on information, contemporary organizations suffer from organizational indifference.**
In our paper the ways and means for achieving Command and Control Information Systems Semantic Interoperability (C2IS) in complex systems are examined, with special emphasis to military systems. The approach within the military domain is presented Multilateral Interoperability Programme (MIP). If implemented correctly and consistently it should lead to achievement of information superiority.

Index Terms - *command and control information system, integration, semantic interoperability.*

I. INTRODUCTION

By following the hypotheses of Peter F. Drucker and Dirk Baecker we can observe the dawn of the so called *next society*. The crucial driving force of the corresponding change is the evolution of computer communication as new media of dissemination: Much more data than ever before can be saved, activated and disseminated. In addition, bits and bytes can't be read like books because they have to be mediated and reproduced by resource consuming software and hardware. Finally computer communication manifests and presses ahead the detemporization, virtualization, and poly-contextuality of information, and is therefore said to be the major driver for an unmanageable complexity of communication [16].

In the military most decision situations possess a degree of ambiguity and uncertainty and are not easily captured in a static or stochastic model. This has led to an increased use of "fuzzy sets" in analyzing decisions, not in an attempt to quantify the unquantifiable, but as a way to formalize our way of dealing with the unquantifiable and imprecise [1], [5]. The concept of fuzziness is related to the idea of the "fog of war" introduced by Carl von Clausewitz. In his discourse *On War*, Clausewitz presents two concepts leading to difficulties in conflict, friction and fog. Friction is the effect of numerous minor incidents which reduce the level of performance so the intended goal is not reached [2]. There are physical and psychological aspects of friction. Friction due to a hostile physical environment is usually more obvious; it is caused by darkness; bad weather or terrain, physical exertion; degraded command and control, logistics, maintenance, or weapon systems; or merely chance bad luck; or psychological factors, such as stress produced by the interaction of combatants and the environment of war. Another source of friction is the "fog of war." Fog is the uncertainty of war, caused by factors such as inaccurate, incomplete or contradictory information, deviations in weapon system efficacy, actions of the enemy, and the enemy's nebulous capabilities and intentions [4]. Operational decision-maker usually faces decisions under conditions of uncertainty-intrinsically imprecise decisions under adverse conditions would normally be faced in military operations.

II. DATA INTEGRATION AND INFORMATION SUPERIORITY

Information superiority is that degree of dominance in the information domain which permits the conduct of operations without effective opposition [18].

Sound thinking and decision making are more than mere loose ends of network-centric warfare. Moreover, success in competition on the military cognitive plane will not necessarily follow success on

the technological plane. Therefore, what is needed is a coherent strategy to build battle-wise forces.

A. Superiority in Decision Making

Decision superiority-"the process of making decisions better and faster than an Adversary"-is essential to executing a strategy based on speed and flexibility. Decision superiority requires new ways of thinking about acquiring, integrating, using and sharing information. It necessitates new ideas for developing architectures for command, control, communications and computers (C4) as well as the intelligence, surveillance and reconnaissance assets that provide knowledge of adversaries. Decision superiority requires precise information of enemy and friendly dispositions, capabilities, and activities, as well as other data relevant to successful campaigns.

Battle space awareness, combined with responsive command and control systems, supports dynamic decision making and turns information superiority into a competitive advantage adversaries cannot match. Persistent surveillance, ISR management, collaborative analysis and on-demand dissemination facilitate battle space awareness. Developing the intelligence products to support this level of awareness requires collection systems and assured access to air, land, sea and space-based sensors.

Decisions to apply force in multiple, widely dispersed locations require highly flexible and adaptive joint command and control processes. Commanders must communicate decisions to subordinates, rapidly develop alternative courses of action, generate required effects, assess results and conduct appropriate follow-on operations. A decision superior joint force must employ decision making processes that allow commanders to attack time-sensitive and time-critical targets. Dynamic decision making brings together organizations, planning processes, technical systems and commensurate authorities that support informed decisions. Such decisions require networked command and control capabilities and a tailored common operating picture of the battle space [18].

Awareness of the growing operational and strategic importance of decision superiority must exist, which has some but not all of the elements of the superiority in cognitive capacity and performance that we call battle-wisdom. It also tells us that responsive command and control systems, collaborative analysis, and on-demand dissemination of information are important to decision superiority. This official explanation of what is required for decision superiority fails to stress human cognition-how people think and how well they decide. It is as if battle-wisdom-the capacity to integrate reliable intuition and rapid reasoning and the abilities to anticipate, decide quickly, seize opportunities, and learn in action-is assumed, needing only better intelligence sensors, information networks, and processes to succeed. It calls for commanders to communicate their decisions to subordinates, without recognizing that the subordinates may well be better informed than their superiors to decide what to do. The networking is not that it enables commanders to promulgate orders but that it informs those "on the edge" and permits them to collaborate, accept responsibility, and take initiative. The key to decision superiority lies not in the information network but in the human [9].

B. Decentralization of Decision making

The value of self-directed learning can be undermined if individuals lack the trust and confidence of their superiors and are not granted the authority to make decisions. Many strong businesses are distributing decision making authority to those on the front lines, a practice that not only enables an organization to act with greater agility and speed but also imparts confidence to those who make the decisions. Businesses in the 1980s and 1990s were swamped with new management theories-to name a few: total quality management, continuous improvement, right-sizing, core competence, process engineering, strategic alliances, competitive strategies, learning organizations, empowerment, flattening of hierarchies, cross-boundary teaming etc.

None of these theories alone induced sustainable organizational change without the mutual commitment of leadership and rank-and-file employees. For reform to be sustainable, an organization must put into practice certain values and principles concerning information, people, and trust: transparency; open information-sharing; cross-boundary communication and collaboration; an understood mission and values; a culture that rewards taking responsibility; a commitment to learning; and a willingness to give talent room and to give people the confidence and authority to make decisions.

Organizations that need to wait for bureaucratic procedures, chain-of-command review, or decisions from on high before acting on an opportunity may not be able to survive in fluid and unfamiliar situations [9].

III. TO BE AWARE OF THE SITUATION

Situation Awareness (SA) has several dimensions and is closely related to Decision Support Systems. SA is the perception of environmental elements within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future. It is also concerned with perception of the environment critical to decision-makers in complex, dynamic areas from aviation, air

traffic control, power plant operations, military command and control (C2). SA involves being aware of what is happening to understand how information, events, and actions will impact goals and objectives, both now and in the near future. It directly depends on Information Superiority and Common Operational Picture (COP). COP is a single identical display of relevant (operational) information (e.g. position of own troops and enemy troops, position and status of important infrastructure such as bridges, roads, etc.) shared by more than one command. A common operational picture facilitates collaborative planning and assists all echelons to achieve situation awareness [17]. Situation awareness may be expressed and presented as follows:

$$SA = f(IS, COP) \quad (1).$$

A. Defence Chief Information Officer's Role and Responsibility

Within defence organizations, situation awareness is generally divided into two categories - real-time battlefield situation awareness (e.g., US Blue Force Tracking) and a broader, less real-time capability often associated with either a balanced scorecard or "dashboard" for senior leadership:

- Battlefield SA - one of the biggest problems the US system has faced is the exponential growth in both usage and data feeds (data from various sensors). The system was not initially designed to support this environment and was developed in a very "stovepipe" fashion. The future of this and similar systems is their ability to adapt quickly to changes in mission, environment, and data streams. The primary role of the Chief Information Officer (CIO) in this environment should be to define and enforce data interoperability standards to enable the system to be agile. Functional proponents and functional oriented developers will seldom see the value or long-term impact of developing to an enterprise vision. This is the prime function a Defence Chief Information Officer (DCIO) can bring to the battlefield systems. However, as further discussed below, linkage of the value of an enterprise ICT approach to mission accomplishment is critical. In the case of battlefield SA, that value is likely best expressed in terms of agility to changing mission situations and interoperability with current and future coalition partners.
- Enterprise SA - in addition to the standards work mentioned above, the DCIO has several additional functions that should be performed in support of this broader Situation Awareness requirement. The most basic function is to work with the mission leads to develop an Information technology (IT) Strategy [7]. Also, key pieces of the IT Strategy are the Enterprise Architecture [15] and the Enterprise Sourcing Strategy [6]. One of the keys to CIO success in transitioning from an infrastructure services operator to a full mission partner is developing a mission performance mindset. Mission performance metrics and how ICT impacts those mission performance metrics is one of the more challenging aspects of CIO life for most defence CIOs, but is also the area that is most powerful in linking ICT capabilities to mission capabilities. The Mission Performance section should be expanded as an Enterprise dashboard for the senior non-IT leadership within the organization.

B. DCIO's Typical challenges

Two main challenges for DCIO exist:

- Credibility - many warfighters and mission leaders have had poor experiences with the stability and responsiveness of central ICT organizations and, therefore, the CIO lacks credibility to "sit at the table". Linking IT performance to Mission Performance metrics is a critical 1st step in establishing that credibility.
- Knowledge - many DCIOs lack the mission understanding to communicate the value of ICT capabilities to the mission in terms the non-ICT leadership understands. There are several ways to overcome this, either by bringing more mission personnel into the ICT organization or detailing a senior ICT manager to work with and even deploy with combat unit to fully understand the mission challenges and vocabulary.

Based on the results of business process analysis, CIO as a leader of ICT function in a complex system can and should initiate process change, streamline the processes or suggest introduction of the new processes if necessary, in order to improve overall business process [8].

Military organizations have been, and will continue to be, challenged to improve the tactics, techniques, and procedures (TTPs) associated with executing their mission. ICT has played a major role in the evolution of TTPs over the past 15 years and that IT role is increasing with the adoption of Net-Enabled Warfare concepts. However, leaving the application of ICT to support business process improvement in the hands of the knowledgeable functional proponent has resulted in numerous marginal systems and an extensive array of stove-piped applications.

Too often, an emerging technology or the latest "gadget" that is purported to solve all their problems overly influences the well-meaning functional lead. Functional proponents, who are paid to be functional experts and not ICT experts, are even more prone to fall into the trap of "peak of inflated expectations". DCIOs have an opportunity and an obligation to take a leading role in the effective and efficient application of ICT to mission process improvement across all functional areas [14].

IV. MULITALATERAL INTEROPERABILITY PROGRAMME

It is necessary to describe Multilateral Interoperability Programme strategy to achieve interoperability of cooperating national Command and Control Information Systems at all levels of command, in support of multinational, combined and joint operations in order to provide a focus for the MIP Programme of work [13].

MIP has a three-tiered structure: two Management levels and one Task-Execution level. The Management levels consist of an oversight Steering group and a Programme Management group. In late 2009 the programme structure was reorganised from Multidisciplinary Working Parties into two Integrated Product Teams (IPT). The first is IPT-3 and it is responsible for the In Service Support to Block 2 and Block 3 MIP products. The second team is IPT- Futures, and it is responsible for the development of a future specification using the latest system architecture concepts.

ACT (Allied Command for Transformation) Members of SEE (Staff Element Europe), DPC (Defence Planning Command & Control) attend the MSG and the PMG meetings. DMSWG (Data Management Services Working Group) cooperates in the development of the main data model called JC3IEDM (Joint Command, Control and Consultation Exchange Data Model) honouring the agreement signed with MIP in February 2004. The JC3IEDM is NATO STANAG 5525. NDAG is under the NATO C3 Board, Subcommittee 5 (Information Services subcommittee) known as ISSC. NC3A (NATO Command, Control and Consultation) implements the MIP specifications in the framework of the current development and deployment [12].

MIP is an interoperability organization established by national C2IS system developers with a requirement to share relevant C2 information in a multinational/coalition environment. As a result of collaboration within the programme, MIP produces a set of specifications which, when implemented by the nations, provide the required interoperability capability.

MIP provides a venue for system level interoperability testing of national MIP implementations as well as providing a forum for exchanging information relevant to national implementation and fielding plans to enable synchronization. MIP is NOT empowered to direct how nations develop their own C2IS.

Key points:

- MIP focuses on interoperability of command and control (C2) systems, which includes the Land view of Joint operations, but encourages contributions from Air, Maritime and other CoIs.
- MIP specifications are based on operational requirements developed into a fieldable interoperability solution.
- MIP assures the quality of the specification through operational and technical testing of national implementations.
-

A conceptual illustration of how the current MIP interoperability solution works is illustrated below (Figure 1).

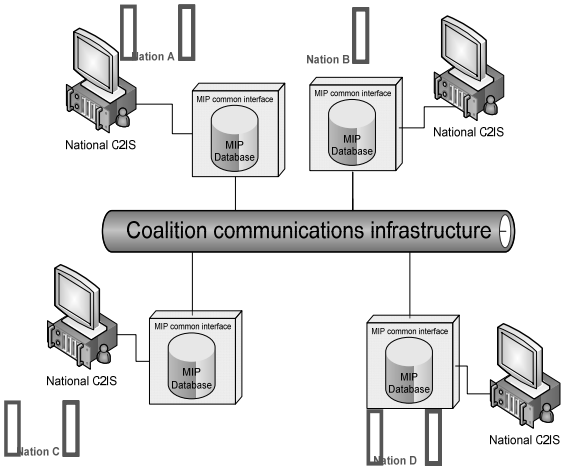


Figure 1: Different coalition C2IS connecting via the current MIP solution

The MIP solution refers to two or more national C2IS exchanging information by employing their respective implementations based upon the agreed MIP technical specifications[1] and supporting procedural and operational documentation.

The vision for the Multilateral Interoperability Programme (MIP) is to become the principal operator-led multinational forum to promote international interoperability of Command and Control Information Systems (C2IS) at all levels of command.

The MIP scope is to deliver a command and control interoperability solution in net centric environment focused on the Land operational user in a Joint environment including the requirements of Maritime and Air communities.
The MIP scope is derived from considering the constraints and limitations of the MIP solution [13].

A. MIP Constraints, limitations and mission

Constraints:

- Resources:
  - Time available to work on the MIP specification.

[1] The technical specification includes a common data model and agreed exchange mechanisms.

- Availability of operational and technical human resources.
- Lack of centralized funding.

- Governance based on consensus.

Limitations:

- A focus on Land interoperability requirements for two reasons:
  - Historical interoperability capability gap in the Land environment.
  - Current operational focus on Land operations.

Mission:
MIP is to further develop and improve interface specifications in order to reduce the interoperability gap between different C2IS.

B. MIP Tasks, main products and documents

- Support fielded MIP solutions.
- Further improve the MIP solution by adopting modern development approaches and standards[2].
- Harmonize with NATO and leverage other appropriate concepts, profiles and standards[3].
- Improve flexibility in using the MIP solution in ad-hoc coalitions.
- Extend the scope of MIP interoperability.
- Engage Maritime, Air and other CoIs to cooperate with MIP.
- Examine better ways of structuring the MIP programme.

The main products from the programme are:
The Joint Consultation, Command & Control Information Exchange Data Model (JC3IEDM) promulgated by NATO as STANAG 5525[4].

Documents:

- Operational documents:
  - Instructions on how to use the MIP solution.
  - Record of incorporated Information Exchange Requirements.
  - The programme's Exchange Mechanism specifications and associated procedures.
- Technical documents: Guidance for nations and CoIs on how to implement the MIP specification within the context of their national C2IS.
- Supporting documents: Procedures for testing the MIP specification.

[2] Examples of approaches and standards include the NATO Architectural Framework (NAF), Model Driven Development, Service Orientation and common standards (XML, UML, RDF, etc.).
[3] Examples include NNEC, APP-11, APP-6, etc.
[4] STANAGs are NATO standardized agreements. STANAG 5525 establishes a common data model that NATO nations individually ratify and implement in their own C2IS.

The programme does not include the following aspects of C2IS development:

- National C2IS hardware and software.
- National C2IS software designed to implement and process the MIP specification. However, MIP provides test events to enable nations to evaluate their own systems against MIP specifications.
- Any responsibility for manning and operating national C2IS [13].

C. Components of MIP

MIP should be understood in the context of its 3 integrated components:

- **Organization –** An international military data interoperability organization that meets to define common information exchange requirements (IERs), which can be exchanged between different national C2IS.
- **Specifications –** Delivering an assured capability for interoperability of information. MIP facilitates interoperability through defining/developing common technical standards and associated documentation. MIP intends to further develop and improve interoperability standards in order to support understanding in a common working environment.
- **Materiel development**
  - A forum for national implementers to synchronize their MIP C2IS material fielding plans.
  - An organization that assists in testing national C2IS in accordance with MIP specifications, which is focused on fielded solutions and iterative development [13].

V. CONCLUSION

In this paper semantic interoperability, along with information superiority as one of the arguments of the situation awareness function is analyzed (besides the common operational picture as another function's argument). Enabled by semantic interoperability some main characteristics of information superiority, common operational picture and situation awareness are elaborated, explained and presented.

Interoperability of information is essential and an assured capability for this is vital. The successful execution of fast moving operations needs an accelerated decision-action cycle, increased tempo of operations, and the ability to conduct operations simultaneously within combined/multinational formations. Commanders require timely and accurate information. Also, supporting command and control (C2) systems need to pass information within and across national and language boundaries. Additionally, forces must interact with non-governmental bodies, and international and national aid organizations. IT must act as a force multiplier to enhance operational effectiveness at each level of command by enabling the sending, receiving,

filtering, fusing, and processing of ever-increasing amount of digital information [11].

Due to complexity of the interoperability solution: process (operational), semantic (systems and data) and technical (computers and networks) author seeks appropriate model for C2IS integration, focusing on semantic interoperabilty. The role of the DCIO along with the challenges he meets in his endeavor to deploy systems which could lead to information superiority is described as well. In effort to examine the possibilities of information superiority achievement, place and role of MIP are investigated, contributing though to the understanding and knowledge about the interoperability approaches to process and semantic interoperability in the military.

The MIP programme is not a formal NATO programme. It is a voluntary and independent activity by the participating nations and organizations [12]. In this connection MIP's approach focused on semantic interoperability of C2ISs, aimed at advancement of Network-Centric Warfare (NCW) which is based on NNEC is presented and shortly described. According to The Institute for Defense & Government Advancement (IDGA) [10], [11] MIP was successful in establishing multinational joint C2 common core data standards that can enable and accelerate achieving transformational data strategies for international and national information sharing. MIP's C2 data sharing standard was adopted in 2007 by NATO, as STANAG 5525.

According to the authors' best knowledge MIP is world's unique approach to interoperability in information superiority achievement and military international cooperation. MIP's approach and solution presents world's military "state of the art" interoperability solution and aims to become the model for information superiority achievement, interconnecting different coalition C2ISs at all levels of command in net centric environment.

ACKNOWLEDGEMENT

This work is the result of the project implementation: Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

REFERENCES

[1] Binaghi E., Rampini A.: Fuzzy decision making in the classification of multisource remote sensing data, Optical Engineering, Vol. 32 No. 6, pp. 1193-1204, Atlanta, USA, 1993.

[2] Clausewitz. C. von: On War, translated by Michael Howard and Peter Paret, Princeton University Press, Princeton, USA, 1989.

[3] Clements S. M.: The One with the Most Information Wins? The Quest for Information Superiority, Graduate School of Logistics and Acquisition Management, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, USA, 1997.

[4] Department of the Air Force (DAF): Basic Aerospace Doctrine of the United States Air Force, Volume II. AFM 1-1. Essay C: Human Factors in War; pp. 17-23, Washington, USA, 1992.

[5] Dockery J. T.: The Use of Fuzzy Sets in the Analysis of Military Command, in *Decision Information*. Ed. Chris P. Tsokos and Robert M. Thrall, Academic Press, New York, USA, 1990.

[6] Dreyfuss C., Maurer W., Cohen L.: Business Value of Services in Sourcing Initiatives, Gartner, Inc. 2008.

[7] Gartner Executive Program: IT Strategy: A CIO Success Kit , Gartner, Inc., USA 2009.

[8] Galinec D., Ferek-Jambrek B. (2008): Process Transformation for Reaching Agility: Chief Information Officer Role, Conference Proceedings, CECIIS 19th International Conference 2008, Faculty of Organization and Informatics, Varaždin, pp. 317-324.

[9] Gompert D. C., Lachow I., Perkins J.: Battle-Wise: Seeking Time-Information Superiority in Networked Warfare, Center for Technology and National Security Policy National Defense University, Washington D.C., USA, 2006.

[10] The Institute for Defense & Government Advancement (IDGA): About IDGA, available at http://www.idga.org/, Accessed: 29th June 2009.

[11] Multilateral Interoperability Programme (MIP): Interoperability, available at `http://www.mip-site.org/`, Accessed: 30th June 2009.

[12] Multilateral Interoperability Programme (MIP): MIP Organisation, available at `https://mipsite.lsec.dnd.ca/Pages/MIPOrganisation.aspx`, Accessed: 27th September 2011.

[13] Multilateral Interoperability Programme (MIP): MIP Vision & SCOPE (MV&S), PMG Edition 6.3, Montebello, Canada, 2009.

[14] Ringdahl B.: Improving Business Processes Executive Summary for Defense Sector - May 2009 EXP Report, Gartner, Inc., USA 2009.

[15] Robertson B.: Enterprise Architecture Research Index: Integrating EA with Business Strategy, Gartner, Inc. 2009.

[16] Swiss Sociological Association: "Identity and organization" Workshop, available at http://www.inderscience.com/mapper.php?id=116, Accessed: 29th June 2009.

[17] US Department of Defense: DOD Dictionary of Military and Associated Terms, Joint Chiefs of Staff (JCS), Washington D.C., USA, 2009.
[18] US Department of Defense: National Military Strategy of the United States, Joint Chiefs of Staff (JCS), Washington D.C., USA, 2004.

Zusammenfassung

In der heutigen Zeit der schnellen Entwicklung von Informations - und Kommunikationstechnologien (ICT), die die Zunahme von der Datenverarbeitung ermöglichen, können mehrere Daten gespeichert, angewendet und/oder vermittelt werden. In der Gegenwart werden die Informationen und die Kenntnisse zur Quelle der strategischen Wichtigkeit in den komplexen Systemen, im Handel oder in der Armee. Andererseits gibt es hier eine Frage, ob die gesammelten Daten glaubhaft sind.Komplexe Systeme müssen mit dem wachsenden Zugang zu den Informationen arbeiten, die nicht so viele Kenntnisse über ihren Ursprung und über ihre Qualität haben. Wegen diesem Informationsmangel an den Informationen leiden die jetzigen Organisationen.

In diesem Artikel werden Verfahren und Mittel geforscht, die zum Erreichen von Command and Control Information Systems Semantic Interoperability (C2IS) dienen. Das soll in den komplexen Systemen mit einer grossen Betonung auf die militärischen Systeme verwirklicht werden. Der Zugang im Rahmen des militärisschen Bereichs wird von Multilateral Interoperability Programme (MIP) präsentiert. Wenn es korrekt und konzistent eingeführt wird, sollte es zum Erreichen von der Informationsüberordnung führen.

An Incremental ASM-based Fuzzy Clustering Algorithm

Radu D. Găceanu and Horia F. Pop
Computer Science Department
BabeŞ-Bolyai University
Cluj-Napoca, Romania
Email: rgaceanu@cs.ubbcluj.ro and hfpop@cs.ubbcluj.ro

***Abstract*—We propose an incremental ASM-based clustering algorithm where the agent movements are governed by fuzzy IF-THEN rules. In the ASM model data items are represented by agents placed in a two dimensional grid. The agents will group themselves into clusters by making simple moves in their environment. They will try to get closer to each other if they are rather similar or to get away from each other if they are rather different. The algorithm allocates a new agent on the grid whenever a new data item arrives. At each step the new agent contacts an agent from the grid and if they are similar then they will group together in the same cluster. Whenever a new cluster is created the agents will try to merge the cluster with one of the previously created clusters. If a newly created agent did not find a similar fellow then it will start an ASM-like process in order to search for one and thus the data is clustered.**

I. Introduction

Several clustering algorithms exist each with its own strengths and weaknesses. Some algorithms need an initial estimation of the number of clusters (k-means, fuzzy c-means); others could often be too slow (agglomerative hierarchical clustering algorithms). Ant-based clustering algorithms often require hybridization with a classical clustering algorithm such as k-means.

In [1] an ant-based clustering algorithm is presented. It is based on the ASM (Ants Sleeping Model) approach. In ASM, an ant has two states on a two-dimensional grid: active state and sleeping state. When the artificial ant's fitness is low, it has a higher probability to wake up and stay in active state. It will thus leave its original position to search for a more secure and comfortable position to sleep. When an ant locates a comfortable and secure position, it has a higher probability to sleep unless the surrounding environment becomes less hospitable and activates it again.

Based on ASM, the authors present an Adaptive Artificial Ants Clustering Algorithm (A^4C) [1] in which each artificial ant is a simple agent representing an individual data object. The whole ant group dynamically self-organizes into distinctive, independent subgroups within which highly similar ants are closely connected. The result of data objects clustering is therefore achieved. However, by using local information only the risk of getting trapped into local optimum solutions exists.

In [2] a Stigmergic Agent System (SAS) combining the strengths of Ant Colony Systems and Multi-Agent Systems concepts is proposed. The agents from the SAS are using both direct and indirect communication. By using direct communication the risk of getting trapped in local optima is lower. However, as showed in [3], most ant-based algorithms can be used only in a first phase of the clustering process because of the high number of clusters that are usually produced. In a second phase a k-means-like algorithm is often used.

In [3], an algorithm in which the behaviour of the artificial ants is governed by fuzzy IF-THEN rules is presented. Like all ant-based clustering algorithms, no initial partitioning of the data is needed, nor should the number of clusters be known in advance. The ants are capable to make their own decisions about picking up items. Hence the two phases of the classical ant-based clustering algorithm are merged into one, and k-means becomes superfluous.

In [4], the clustering problem is approached by the idea of context-aware ASM agents. The agents are able to detect changes in the environment and adjust their moves accordingly. The advantage of this approach is that it enables the ants to communicate directly like in [2] therefore breaking the neighbourhood boundaries and thus decreasing the chance of ants to get trapped in local minima. The fuzzy IF-THEN rules governing the agents' movements are also allowing the agents to go beyond the neighbourhood limits; for example in the case of two very different (VD) agents it makes no point to keep them in a reachability distance; and it makes sense that two very different (VD) agents should move further away from each other than two different (D) agents do. Thus the system behaves more naturally. The agents are able to adapt their movements if changes in the environment would occur and this is an important feature in real-time systems, wireless sensor networks or data streams.

We propose an incremental algorithm based on the ASM (Ants Sleeping Model) [1], [4] in order to resolve the clustering problem. Incremental clustering is used to process sequential, continuous data flows or data streams and in situations in which cluster shapes change over time. They are well fitted in real-time systems, wireless sensor networks or data streams because in such systems it is difficult to store the datasets in memory. Incremental clustering algorithms in general do not rely on the in-memory dataset and therefore the space and also time requirements for such algorithms are small.

The rest of the paper is structured as follows. In Section II the theoretical background is presented. The proposed model

is described in Section III and Section IV contains a case study. The advantages and drawbacks of the approach together with some concluding remarks are presented in the closing Section V.

II. Theoretical background

In machine learning, clustering is an example of unsupervised learning because it does not rely on predefined classes and class-labelled training examples. So it could be said that clustering is a form of learning by observation, rather than learning by examples. In data analysis, efforts have been conducted on finding methods for efficient and effective cluster analysis in large databases. The main requirements for a good clustering algorithm would be the scalability of the method, its effectiveness for clustering complex shapes and types of data, dealing with high-dimensional data, and handling mixed numerical and categorical data in large databases.

Fuzzy logic can be viewed as an extension of the Boolean logic. It's important to note that basically any theory could be fuzzified by replacing the classical sets with fuzzy sets. Fuzzy sets could be applied in modelling inexact behaviour, which was not very convenient via the classical set theory. While the classical set theory deals with well defined objects, the fuzzy set theory deals with objects which have a certain membership degree.

There are many applications of fuzzy logic, but actually "is there need for fuzzy logic"? This is the question that Zadeh addresses in [5]. There were and maybe still are many people who are reluctant to fuzzy logic. As it is very well explained, fuzzy logic is not fuzzy, it is actually a precise logic of imprecision and approximate reasoning. It is concluded that the progress from bivalent to fuzzy logic is a natural step forward and an important evolution of science. This is because the real world is a fuzzy one so in order to deal with a fuzzy reality, fuzzy logic is needed. According to Zadeh, in coming years, fuzzy logic is likely to grow in acceptance.

An agent is an entity that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors [6], [7]. An agent that always tries to optimize an appropriate performance measure is called a rational agent. Such a definition of a rational agent is fairly general and can include human agents (having eyes as sensors, hands as effectors), robotic agents (having cameras as sensors, wheels as effectors), or software agents (having a graphical user interface as sensor and as effector).

According to [6], [7] agents exhibit the following characteristics: autonomy, reactivity, pro-activity, sociability, intelligence, mobility, self-organization.

Usually agents coexist and interact forming Multi-agent Systems (MAS). In computer science, a MAS is a system composed of several interacting agents, collectively capable of reaching goals that are difficult to achieve by an individual agent or monolithic system.

The ACO (Ant Colony Optimization) metaheuristic is composed of different algorithms in which several cooperative agent populations try to simulate real ants behaviour. Initially ants wander randomly in order to find food, but they leave pheromone trails in their way. If another ant finds the trail it will likely follow it rather than continue its random path, thus reinforcing the trail.

In ASM (Ants Sleeping Model), an ant has two states on a two-dimensional grid: active state and sleeping state. When the artificial ant's fitness is low, it has a higher probability to wake up and stay in active state. It will thus leave its original position to search for a more secure and comfortable position to sleep. When an ant locates a comfortable and secure position, it has a higher probability to sleep unless the surrounding environment becomes less hospitable and activates it again. Since each individual ant uses only a little local information to decide whether to be in active state or sleeping state, the whole ant group dynamically self-organizes into distinctive, independent subgroups within which highly similar ants are closely connected. The result of data objects clustering is therefore achieved.

III. Incremental ASM clustering

The idea behind incremental clustering is that it is possible to consider one instance at a time and assign it to existing clusters without significantly affecting the already existing structures. Only the cluster representations need to be kept in memory so not the entire dataset and thus the space requirements for such an algorithm are very small. Whenever a new instance is considered an incremental clustering algorithm would basically try to assign it to one of the already exiting clusters. Such a process is not very complex and therefore the time requirements for an incremental clustering algorithm are also small.

Our incremental clustering approach is based on the ASM-like algorithm from [1]. In the ASM model, due to the need for security, the ants are constantly choosing a more comfortable environment to sleep in. The ants feel comfortable among individuals having similar characteristics. In ASM, each data item is represented by an agent, and his purpose is to search for a comfortable position for sleeping in his surrounding environment. While he doesn't find a suitable position to have a rest, he will actively move around to search for it and stop when he finds one; when he is not satisfied with his current position, he becomes active again. The definitions 1-5 from bellow are taken from [1]:

Definition 1.
Let G represent a two-dimensional array of all positions $(x, y)\epsilon[0..2\lceil\sqrt{n}\rceil - 1]^2$, where $G(x, y)\epsilon Z^+ \bigcup\{0\}$ and n is the number of agents. The size of the grid depends on the number of agents in order to avoid overcrowding the grid or, conversely, allocating a grid which consumes too many resources.

The grid in ASM is similar to that of cellular automata. G(x, y) = i, if there is an agent labeled i at position (x, y), otherwise G(x, y) = 0. Unless otherwise stated, the variables

used here are all integers. The ASM uses a grid topologically equivalent to a sphere grid.

The advantages of this grid are, on the one hand, it can ensure the equality of all the locations in the grid (so the cells from the margins have the same number of neighbours as the cells from the interior of the grid i.e. one can jump from one of the northmost positions to one of the southmost positions with one step). On the other hand, the operation style of cellular automata can be borrowed to expedite the agents' movement and computation.

Definition 2.
Let an agent represent a data object by using $agent_i$ to represent the i^{th} agent, and n be the number of agents. The position of an agent is represented by (x_i, y_i), namely $G(agent_i) = G(x_i, y_i) = i$ In ASM, each agent represents one data object.

Definition 3.
$N(agent_i) = N(x_i, y_i) = \{(x, y) mod(2\lceil\sqrt{n}\rceil)| \mid x - x_i \mid \leq s_x, \mid y - y_i \mid \leq s_y\}$
$L(agent_i) = L(x_i, y_i) = \{(x, y)|(x, y) \epsilon N(agent_i),$
$G(x, y) = 0\}$

In the formulas from above, $N(agent_i)$ represents the neighbourhood of $agent_i$, $L(agent_i)$ is the set of empty positions in the neighbourhood, s_x and s_y are the vision limits in the horizontal and vertical direction respectively.

Definition 4.
$f(agent_i) = \frac{1}{v} \sum_{agent_j \epsilon N(agent_i)} \frac{\alpha^2}{\alpha^2 + d(agent_i, agent_j)^2}$,
$\alpha = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} d(agent_i, agent_j)$,
$v = (2s_x + 1)(2s_y + 1)$,
$f(agent_i)$ represents the current fitness of agent i. α is the average distance between the agents. The distance between two agents, $d(agent_i, agent_j)$, is the Euclidean distance between the two agents from the grid.

Definition 5.
$p_a(agent_i) = cos^{\lambda}(\frac{\pi}{2} f(agent_i))$
In the above, $\lambda \epsilon R^+$ is a parameter, and can be called agents' activation pressure. Function $p_a(agent_i)$ represents the probability of the activation of the agent by the surroundings.

If the fitness is low then the probability of activation is high so the agent is probably going to wake up, move and search for a better place to sleep. Conversely if the fitness is high then the probability is low so most probably the agent will stay and sleep.

At the beginning of the algorithm from [1], the agents are randomly scattered on the grid in active state. They randomly move on the grid. In each loop, after the agent moves to a new position, it will recalculate its current fitness f and probability p_a so as to decide whether it needs to continue moving.

If the current p_a is small, the agent has a lower probability of continuing moving and higher probability of taking a rest at its current position. Otherwise the agent will stay in active state and continue moving. The agent's fitness is related to its similarity with other agents in its neighbourhood.

With increasing number of iterations, such movements gradually increase, eventually, making similar agents gathered within a small area and different types of agents located in separated areas. Thus, the corresponding data items are clustered.

Definition 6.
We use the following definition for the fitness in this paper:
$f(agent_i) = \frac{1}{v} \sum_{a_j \epsilon N(a_i)} \frac{\alpha^2}{\alpha^2 + disim(a_i, a_j) de(a_i, a_j)}$,
a_i and a_j respectively denote $agent_i$ and $agent_j$,
$de(a_i, a_j)$ represents the Euclidean distance between the agents,
$disim(a_i, a_j)$ denotes the dissimilarity between the two agents. It could be the Euclidean distance between the agents.

A high level pseudo-code of our incremental clustering algorithm is presented bellow.

```
Algorithm Clustering is
    initialize parameters α, λ, t, s_x, s_y
    initialize an agent a_0 for the first arrived item and
        create a cluster c_0 containing agent a_0
    while (condition)
        for each item i
            create an agent a_i and randomly place it on the grid
            j ← random{0, i}
            if isSimilar(a_i, a_j)
            then
                groupSimilar(a_i, a_j)
            else
                add(U, a_i) // U — the set of unclustered agents
            endif
            if hasElements(U)
            then
                a_asm ← getRandomAgent(U)
                tryActivate(a_asm)
            endif
        endfor
        adaptively update parameters
    endwhile
endAlgorithm
```

The algorithm continuously receives new items to be clustered. For the first such item, a new agent is created that encapsulates this item. As in the classical ASM model, one agent per item will be allocated. A new cluster is created (the first one) and the agent is added to this cluster. In the following, whenever a new item i arrives, a new agent, a_i, is created. This agent is randomly placed on the grid and contacts an already existing agent from the grid. If they are similar then they are grouped together otherwise the agent a_i is added to the set of unclustered agents, U. The $groupSimilar(a_i, a_j)$ procedure groups together two agents, i.e., the agent a_i moves towards the agent a_j on the grid and also it is added to the a_j's cluster. If a_j is not already in a cluster then a new cluster containing both agents is created. Whenever a new cluster is

created we try to merge it with an already existing cluster. The merging is performed based on the similarity between the cluster representatives. The representative of a cluster is either the first item that was added to the cluster or an item pointed out, i.e., manually chosen by the data analyst. The similarity between two agents will be discussed immediately. To the cluster representatives we will come back in the experiments section.

In the final step of the for loop we test if the set of unclustered agents is not empty and if it contains elements then we try to activate a randomly extracted agent from there. In $tryActivate(a_{asm})$ the agent will follow an ASM-like clustering process similar to the one from [4]. If an agent finds a similar fellow then it will call the $groupSimilar(a_i, a_j)$ procedure.

In [4], the agents decide upon the way they move on the grid according to their similarity with the neighbours, using fuzzy IF-THEN rules. Thus two agents can be similar (S), different (D), very different (VD). If two agents are similar or very similar they would get closer to each other. If they are different or very different they will get away from each other. The number of steps they do each time they move depend on the similarity level. So if the agents are VD they would jump many steps away from each other; if they are D they would jump less steps away from each other. In the end the ants which are S will be in the same cluster. The similarity computation is taking into account the actual structure of the data or the data density from the agent's neighbourhood; a bigger change from one agent to another translates into a certain similarity which then affects the agent's movement on the grid. A graphical representation of a fuzzy variable $Similarity$ is shown in figure 1.

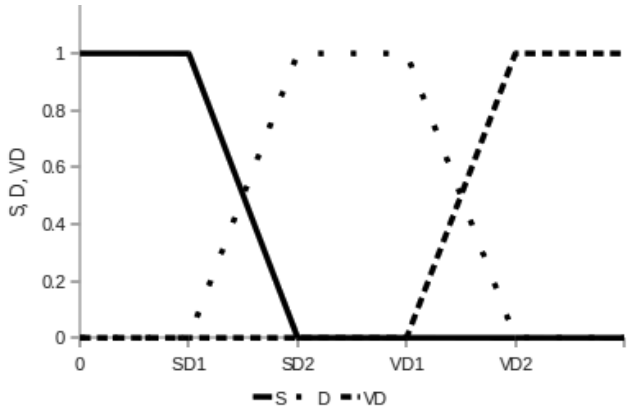


Fig. 1. The fuzzy sets S, D and VD corresponding respectively to the linguistic concepts $Similar$, $Different$ and $VeryDifferent$ are called the $states$ of the fuzzy variable $Similarity$. The limits $SD1$, $SD2$, $VD1$, $VD2$ are application specific.

The parameter α is the average distance between agents and this changes at each step further influencing the fitness function. The parameter λ influences the agents' activation pressure and it may decrease over time. The parameter t is used for the termination condition which could be something like $t < t_{max}$. The parameters s_x, s_y, the agent's vision limits may also be updated in some situations.

IV. Experiments

In order to test the algorithm in a real-world scenario, the Iris dataset [8] was considered for a first test case. The data set contains 3 classes of 50 instances each, each class referring to a type of iris plant. There are 4 attributes plus the class: sepal length in cm, sepal width in cm, petal length in cm, petal width in cm, class (Iris Setosa, Iris Versicolour, Iris Virginica). This dataset is appropriate for rather testing classification, but it was preferred for clustering too because the class attribute is given and hence there is a way to evaluate the algorithm. So apparently it would be ideal for the algorithm to produce 3 clusters of 50 instances, the 3 clusters corresponding to the given 3 classes.

The last 2 attributes (petal length in cm and petal width in cm) are highly correlated according to [8]. However we do not dismiss any of these attributes because we would like to keep as much of the data unchanged. We do however scale the data to the interval $[0, 1]$. At this point the clustering process can be started. In the final grid configuration one would clearly see the clustered agents. Due to space limitations we can't show here the final grid configuration, but we will immediately summarize the results. Besides the final grid configuration a membership table is also produced. The membership table shows the membership degree of each agent to the clusters. Relevant parts of the membership table will also be explained immediately.

According to the Iris dataset [8], items ranging from 0 to 49 belong to the first class, items ranging from 50 to 99 belong to the second class and items ranging from 100 to 149 belong to the third class. So from the grid table it appears that the following clusters contain some misclassifications:

- $Cluster1$ (items 0 – 49): no misclassifications
- $Cluster2$ (items 50 – 99): 106, 119, 133, 134
- $Cluster3$ (items 100 – 149): 70, 77

So it appears that the algorithm has misclassified 6 items.

Let us check each item in more detail. In order to do this deeper analysis we have to check out the membership degrees table. In the membership table one can see in what degree does each item belong to the clusters. This membership is computed with respect to the cluster representatives. The representative of a cluster is either the first item from that cluster or an item designated by the data analyst. In case of cluster merging if any of the representatives is designated by data analyst then this will be the representative of the new cluster; otherwise the representative of the cluster with the greatest number of elements will be chosen as the new representative. The agents 45, 95, 145 resulted from this process as final representatives for clusters 1, 2 and 3 respectively.

In the following, the similarity between each of the above reported misclassification and the corresponding representative is given. The membership degree to each cluster is also considered. The Euclidean distance between items has been

used as a similarity metric. So a small similarity value i.e. distance between two items means that the two items are similar, should stay together.

MisclassificationId	Similarity	C1	C2	C3
106	0.25	0	0.95	0.84
119	0.24	0	0.96	0.83
133	0.19	0	1	0.89
134	0.24	0	0.96	0.82

TABLE I
CLUSTER2 — REPRESENTATIVEID (95)

From the table I it can be seen that the similarities between the considered items and the representative lie between 0.19 and 0.25. This makes them S ($Similar$) with this item. The membership degree with $Cluster2$ suggests that these items belong to this cluster. However the membership degree with $Cluster3$ is also high. The highest membership degree is with $Cluster2$ though and because of this it could be claimed that the items are actually correctly classified with respect to the considered metric. However we believe that these items cannot be considered to strictly belong either to $Cluster2$ or to $Cluster3$ as they are clearly at the border of the two clusters so they belong to both. In this case we also believe that they should not be regarded as misclassifications.

Let us check the reported misclassifications from the third cluster.

MisclassificationId	Similarity	C1	C2	C3
70	0.22	0.0	0.94	0.98
77	0.24	0.0	0.95	0.96

TABLE II
CLUSTER3 — REPRESENTATIVEID (145)

Like the items from table I, the items from the table II should also not count as misclassifications for the same reasons from above. So, arguably, the algorithm has a 100% classification accuracy for the considered dataset.

It may be observed that starting with item 50 a lot of instances have high membership degrees to both $Cluster2$ and $Cluster3$. This suggests that the members of these two classes are not linearly separable as mentioned in [8] so a clustering process could also merge these two clusters and hence report only two clusters instead of three.

For the second case study the wine dataset [9] was considered. This dataset contains the results of a chemical analysis of wines grown in the same region in Italy but derived from three different wine growers. The analysis determined the quantities of 13 constituents found in each of the three types of wines. In [9] it is mentioned that the initial dataset had 30 attributes. So the current dataset has 13 attributes plus the class. There are 178 instances grouped in three classes corresponding to the three wine growers. Items ranging from 1 to 59 belong to the first class, items from 60 to 130 belong to the second class and items from 131 to 178 belong to the third class.

The same procedure as the one form the first case study was applied and the following misclassifications resulted from the final grid configuration:

- $Cluster1$ (items 0 – 58): 63, 65, 66, 73, 78, 95, 98
- $Cluster2$ (items 59 – 129): 130, 134
- $Cluster3$ (items 130 – 177): 83

From the above it appears that the algorithm has misclassified 10 items. Moreover, the following items have not been classified, i.e., they remained in the collection U of unclustered agents: 59, 110, 121, 123, 124. So it appears that there are 15 classification errors. But let us check out everything in more detail.

The agents 17, 89, 166 are the final representatives for clusters 1, 2 and 3 respectively.

In the following, the similarity between each of the above reported misclassification and the corresponding representative is given. The membership degree to each cluster is also considered. The Euclidean distance between items has been used as a similarity metric. So a small similarity value i.e. distance between two items means that the two items are similar, should stay together.

MisclassificationId	Similarity	C1	C2	C3
63	0.63	0.87	0.83	0
65	0.42	1	0.99	0
66	0.64	0.86	0.83	0
73	0.61	0.89	0	0
78	0.69	0.81	0	0
95	0.69	0.81	0	0
98	0.51	0.99	0.8	0

TABLE III
CLUSTER1 — REPRESENTATIVEID (17)

From the table III it can be seen that the similarities between the considered items and the representative are bellow 0.7. This makes them still S ($Similar$) with this item. The membership degree with $Cluster1$ suggests that these items belong to this cluster. However the membership degree with $Cluster2$ is also high. The highest membership degree is with $Cluster1$ though and because of this it could be claimed that the items are actually correctly classified with respect to the considered metric.

Let us check the reported misclassifications from the second cluster.

Cluster2 — RepresentativeId (89)				
MisclassificationId	Similarity	C1	C2	C3
130	0.76	0	0	0
134	0.68	0	0.82	0.8

TABLE IV
CLUSTER2 — REPRESENTATIVEID (89)

In table IV, it can be seen that item 134 has a high membership degree to both $Cluster2$ and $Cluster3$. Again, the highest membership is with $Cluster2$, the cluster in which

it was classified. So it could be considered that it is correctly classified with respect to the considered metric. According to the same criterion, item 130 should not belong to any of the clusters, it should be labelled as an outlier. However it is incorrectly classified to $Cluster2$.

Let us check the reported misclassifications from the third cluster.

MisclassificationId	Similarity	C1	C2	C3
83	0.6	0	0.81	0.9

TABLE V
CLUSTER3 — REPRESENTATIVEID (166)

In table V, it can be seen that item 83 has a high membership degree to both $Cluster2$ and $Cluster3$, but the highest membership is with $Cluster3$, the cluster in which it was classified. Again it could be considered that it is correctly classified with respect to the considered metric. So up to this point only the item 130 was incorrectly classified to $Cluster2$.

Let us now analyse the items from the collection U of unclustered items.

ItemId	SimC1	C1	Sim C2	C2	Sim C3	C3
59	1.06	0	0.75	0	1.08	0
110	0.91	0	0.93	0	1.11	0
121	0.73	0	0.96	0	1.2	0
123	1	0	0.9	0	0.99	0
124	0.94	0	0.86	0	1.13	0

TABLE VI
UNCLUSTERED ITEMS

As it can be seen from the Unclustered items table, all the elements from the collection U have very high similarity values with respect to each cluster and this implies membership degrees equal to zero. So these items are correctly left apart of any cluster.

Consequently, it can be argued that only item 130 is truly a misclassification with respect to the considered metric. On the other hand, it is clear that the approach has 15 classification errors if the results from [9] were to be taken ad litteram. This may suggest that another metric should be considered for computing the similarity between items. From a classification point of view, even in the most pessimistic result interpretation (15 classification errors), the accuracy would be 91%.

Bottom line: we have seen in both tests that most of the apparently classification errors were actually items that have high membership degrees to more than one cluster. Nevertheless, in our opinion, it is clear that we are dealing with hybrid data. Actually the hybrid nature of the data is suggested in [8] where it is stated that one class is linearly separable from the other two, but the latter are not linearly separable from each other. By using fuzzy methods such features of the data are easy to be observed. We have also discovered hybrid data in the case of the wine dataset [9]. In this case we assume that the quality of data is not good enough in order for clearly separate between different wine breeds. This idea is strengthened by [9] where it is stated that the dataset initially had around 30 variables, but the current one only has 13 variables. So it is unimportant if in our experiments we have obtained a certain number of classification errors with respect to the results from [8], [9] as long as our approach allows us to have an insight into these data such that we can actually see that we deal with hybrid items.

V. CONCLUSION

The algorithm we have presented is based on the adaptive ASM approach from [1]. When being in ASM mode, the agent movements are following fuzzy IF-THEN rules for deciding upon the direction and length of the movement like in [4]. The major difference is that the clustering approach is an incremental one. Incremental clustering is used to process sequential, continuous data flows or data streams and in situations in which cluster shapes change over time. They are well fitted in real-time systems, wireless sensor networks or data streams because in such systems it is difficult to store the datasets in memory. The algorithm considers one instance at a time and it basically tries to assign it to one of the existing clusters. Only cluster representations need to be kept in memory so computation is both fast and memory friendly. Experiments on real-world datasets [8], [9] show good clustering results and suggest that the method is a promising one. The fuzziness of the approach allows the discovery of hybrid items in both datasets [8], [9]; the fact that there are hybrid items could be an indication of the quality of data. More experiments with other clustering methods using larger, real-world data sets are on-going.

ACKNOWLEDGMENT

The authors wish to thank for the financial support provided from programs co-financed by The Sectorial Operational Programme Human Resources Development, Contract POSDRU 6/1.5/S/3 "Doctoral studies: through science towards society".

REFERENCES

[1] L. Chen, X. H. Xu, and Y. X. Chen, "An adaptive ant colony clustering algorithm," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, Vol. 3*, 2004, pp. 1387–1392.
[2] C. Chira, D. Dumitrescu, and R. D. Găceanu, "Stigmergic agent systems for solving NP-hard problems," *Studia Informatica*, vol. Special Issue KEPT-2007: Knowledge Engineering: Principles and Techniques (June 2007), pp. 177–184, June 2007.
[3] S. Schockaert, M. D. Cock, C. Cornelis, and E. E. Kerre, "Fuzzy ant based clustering," in *Ant Colony Optimization and Swarm Intelligence, 4th International Workshop (ANTS 2004), LNCS 3172*, 2004, pp. 342–349.
[4] R. D. Găceanu and H. F. Pop, "A context-aware ASM-based clustering algorithm," *Studia Universitatis Babes-Bolyai Series Informatica*, vol. LVI, no. 2, pp. 55–61, 2011.
[5] L. A. Zadeh, "Is there a need for fuzzy logic," *Information Sciences*, vol. 178, no. 13, pp. 2751–2779, July 2008.
[6] G. Serban, *Sisteme Mutiagent in inteligenta artificiala distribuita. Arhitecturi si aplicatii*. Cluj-Napoca: Ed. Risoprint, 2006.
[7] G. Serban and H. F. Pop, *Tehnici de Inteligenta Artificiala. Abordari bazate pe Agenti Inteligenti*. Cluj-Napoca: Ed. Mediamira, 2004.
[8] "http://archive.ics.uci.edu/ml/datasets/iris."
[9] "http://archive.ics.uci.edu/ml/datasets/wine."

Valuation of Business in Virtual Reality

Paweł Kossecki
The Lodz Filmschool
ul. Targowa 61/63
90-323 Łódź
Poland
kossecki@poczta.onet.pl

***Abstract* — Recent years brought fast growth and fall of popularity of virtual worlds and virtual ventures. Many companies moved their activity to the virtual reality. Virtual worlds strongly increased their communities. Researches show that they influence on the real world activity. Selected problems related to valuation and defining of economic activity in virtual reality were described. Author has developed a model of business valuation in virtual reality, based on the Customer Lifetime Value method.**

Keywords: Internet, Valuation, Virtual Reality, Second Life

I. INTRODUCTION

Recent years brought fast growth and fall of popularity of virtual worlds and virtual businesses. Many companies moved their activity to the virtual reality. Virtual worlds strongly increased their communities. *Masive Multiplayer Online Games (MMOGs)* increased their popularity. The most famous virtual world is launched in 2003 by Linden Lab *Second Life*. Residents of *Second Life* can buy or rent virtual land, create content of virtual world and buy virtual products. They can trade with each other by using virtual currency called Linden dollars, which can be exchanged to real US dollars on the virtual currency exchange market. Apart from the well known e-commerce, there is a noticed development of the virtual commerce (*v-commerce*).

TABLE I. COMMERCE CATEGORIES [1]

Commerce Type	Location	Customer Type	Income Type
Virtual Commerce	Virtual Store in a Virtual World	Avatar	Virtual and/or Real Currency
Electronic Commerce	Webstore, or Conventional E-Commerce Ready Website	Internet Shopper	Real Currency
Physical Commerce	Brick-and-Mortar Store	Physical Person	Real Currency

V-commerce ventures grow fast but may also fall in a short time. Many virtual stores are empty [13]. Researches show that they influence on the real world activity. Many real companies started to operate in the virtual reality environment. Some of them just to increase brand awareness of real products, some of them heavily invested to create virtual stores. Virtual residents are called avatars. Avatars can interact with other avatars, wear virtual apparels, interact with virtual products, use virtual services, purchase them. They can buy virtual world versions of real life products, or specially designed products and services for the virtual world environment, like educational services.

Virtual worlds can be used in three basic areas [19]:

- Collecting orders – client can make order for a product, which will be delivered physically (apparels) or virtualy (law advice).
- Client service – virtual world can be used as information center for clients.
- Probationary or real virtual consumption – virtual world can be used to demonstrate products available on virtual or real market. For example avatar can enjoy ride by a new model of a car or wear virtual apparels.

TABLE II. MOTIVATIONS OF POLISH USERS OF SECOND LIFE [19]

What's most important in Second Life? (max. 2 answers)	% of users indicating answer
Make friends	76%
Earning money	38%
Building objects in virtual world	30%
Education	18%
Other	42%

38% of polish virtual world users indicate earning money as the most important motivation for using Second Life. Virtual 3D shops are more natural shopping environment in comparision to traditional e-commerce sites. They can offer more satisfaction from purchasing. Positive correlation between experience of 3D virtual environment (*telepresence*) and positive attitude to the products is to be observed [10].

There are following aspects of using virtual worlds in the selling process and building brand awareness [18]:

- Free contact with the client.
- Building trust to the supplier.
- Possibility of using virtual CRM.

- Safety of personal data.
- Lack of budget restrictions (virtual products are cheap in production; there is no incremental cost related to production of a next copy).

As in case of any new business, there is a question how to value those new, high growth virtual businesses?

Valuation of a high growth business is always complex. Most of the new, high growing businesses in virtual reality have negative cash flows. These businesses have often limited history to draw upon for future projections. They also typically invest heavily in the early periods of their activity, resulting in negative cash flows. Consequently, traditional financial methods have difficulties in valuating these businesses. It is hard to use a P/E (price to earning) ratio for a company that has no or negative earnings, or to use the Discounted Cash Flow approach when a firm has negative cash flow.

The model described below is an enhancement of the model proposed by Kossecki for valuation of Internet companies, based on Customer Lifetime Value (CLV) and partition of customers into one-time deal clients and those oriented to creating relationships with the suppliers [11], [12]. Number of customers was used in prior analyses of the traditional internet market as a main indicator of the company's value. In 1999, when the Internet market flourished, valuing companies was based on volume of the customers' base, without considering their potential in generating profits, e.g. one customer of the Internet bookstore Amazon was valued $3000, whereas one customer of the protal Yahoo! was valued up to $1000 [8].

The simplest formula for calculating CLV is the following:

$$V_{CLV} = n\sum_{t=0}^{CL}\frac{P_t}{(1+r)^t} \tag{1}$$

V_{CLV} – value of an enterprise based on customer lifetime value
n – number of customers
P_t –profit per individual user in t-period
CL – customer lifetime
r – discount rate

The easiest way to forecast number of customers,is to use forecasts of the company management. If using models is preferred, it can be estimated based on bass model [3], which is quite complicated, or logistic curve, according to the following equation:

$$N_t = \frac{a}{1+e^{-b-ct}} \tag{2}$$

N_t – number of customers in the t – period of time
a, b, c – function parameters

Number of new customers is to be calculated according to the following formula:

$$n_t = \frac{dN_t}{dt} = \frac{ace^{-b-ct}}{(1+e^{-b-ct})^2} \tag{3}$$

Parameter a of the logistic curve describes the point of absorption and $a/2$ shows the point of inflection. Methods of estimating parameters of the logistic curve are described e.g. by Stanisz [16]. Customer retention or churn rates should be known [11], [12].

A new client is profitable only then if the future discounted cash flows are higher than the acquisition cost.

II. MODEL

In the literature, there is lack of the appropriate measures that would allow for value assessment of virtual world business ventures [2]. Problem of valuation in virtual reality and business value creation was analyzed only by few researchers [1].

Author will concentrate on the virtual commerce type of virtual ventures. Virtual consumer - '*avatar*' - purchases products in virtual store located in virtual world by using virtual currency.

Following sources of value for owners of the virtual commerce are to be considered:

- Selling of virtual products.
- Selling of real products.
- Marketing – possesing virtual business may influence on brand awareness and sales of real products in real world.
- Advertisement revenues.

In contrast to the previous papers of Author [11], [12], where several types of customers were considered, in case of virtual world there is no necessity to do that. There will be only two types of ventures: B2C and B2B. Both of them will have similar sources of revenues and one group of clients.

Value of a virtual enterprise V_{CLV} can be calculated according to the following formula:

$$V_{CLV} = n(V_{AR} + V_{VP} + V_{RP} + P_S - SAC) \tag{4}$$

V_{AR} – advertisement incomes
V_{VP} – virtual products incomes
V_{RP} – real products incomes
P_S – value of real options
SAC - customer acquisition cost

Advertisement incomes V_{AR} can be calculated according to the following formula:

$$V_{AR} = \sum_{t=0}^{EL} \frac{i_t * i_{V,t} * P_{A,t}}{(1+r)^t} \quad (5)$$

i_t — average number of views related to the visit of a customer
$i_{V,t}$ — average number of visits of a customer
$P_{A,t}$ — advertisement profit per single view
EL – time horizon for estimating CLV

Virtual products incomes V_{VP} can be calculated according to the formula:

$$V_{VP} = \sum_{t=0}^{EL} \frac{i_{VP,t} * P_{VP,t}}{(1+r)^t} \quad (6)$$

$i_{VP,t}$ — number of virtual products sold to a customer
$P_{VP,t}$ — average margin from the virtual product sale

Real products incomes V_{RP} can be described by the following formula:

$$V_{RP} = \sum_{t=0}^{EL} \frac{i_{RP,t} * P_{RP,t}}{(1+r)^t} \quad (7)$$

$i_{RP,t}$ — number of real products sold
$P_{RP,t}$ — average margin from the real product sale

Only one kind of virtual and real products is considered. If there are more kinds of products, appropriate summation is to be introduced. Values of real options can be calculated according to the binomial or Black-Scholes formula [4], which is considered as the best tool for valuation it [17].

Price of a *call option*, according to the modified Black-Scholes formula, in the case of an european type option (European type options can only be exercised on the expiration date) is represented by the following formula:

$$P_c = V_P * N(d_1) - I * e^{-r_{RF} * t_e} N(d_2) \quad (8)$$

P_c — call option price
V_P — discounted cash flow generated by the investment project
N — standard normal cumulative distribution function
r_{RF} - risk-free interest rate
t_e — time remaining until expiration of option
I — strike price

$$d_1 = \frac{\ln \frac{V_P}{I} + \left(r_{RF} + \frac{\sigma^2}{2} \right) t_e}{\sigma \sqrt{t_e}}$$

$$d_2 = d_1 - \sigma \sqrt{t_e}$$

σ — volatility of discounted cash flow generated by the project

Time remaining until expiration of option is the period of holding decision regarding the investment project, strike price is the cost of expenses related to the project.
Price of a *put option* is represented by the equation shown below:

$$P_P = -V_P * N(-d_1) + I * e^{-r_{RF} * t_e} N(-d_2) \quad (9)$$

P_P — put option price

Black-Scholes formula is based on the following assumptions:

- Returns on the underlying price of an instrument are lognormally distributed.
- No taxes, commissions and transaction costs.
- No arbitrage.
- Risk free rate is constant over time.
- Liquidity – market is efficient, it is possible to purchase or sell any amount of options at any given time.
- Volatility of discounted cash flow is constant [7].

Assumptions, advantages and limitations of the Black-Scholes formula are described by many researchers [9], [16]. Other tool for valuation of real options is the binomial model. Binomial model has similar assumptions to Black-Scholes formula [15]. It was for the first time presented by Cox, Ross and Rubinstein [5], and was widely described by many researchers [6], [14], [15].

The binomial model divides the time to expiration into number of time intervals. A tree is initially produced working forward from the present to expiration. At each step it is assumed that the stock price (discounted cash flow of the investment project) will move up or down by an amount calculated using volatility and time to expiration. This produces a binomial distribution tree, which represents all the possible paths that the stock price could take during the life of an option.

III. CONCLUSIONS

It is difficult to properly valuate and predict the future of a business in the virtual world, however the model described above can be used by a virtual enterpreneur to valuate his business in a virtual reality and predict probability of surviving in this volatile environment. This model can be used also for valuating small e-commerce shops. It allows to valuate business, as well as to look inside a company and understand where its value is created.

It is possible to further enhance the model by introducing several groups of clients and products, calculating cash flows and incomes. The model doesn't include advantages directly related to creating brand awareness in a virtual reality.

REFERENCES

[1] R. Y. Arakji, K. R. Lang, "Avatar business value analysis: a method for the evaluation of business value creation in virtual commerce," Journal of Electronic Commerce Research, vol. 9(3), 2008.

[2] S. Barnes, "Virtual worlds as a medium for advertising," The DATABASE for Advances in Information Systems, vol. 38(4), pp. 45-55, 2007.

[3] F. M. Bass, "A new product growth model for consumer durables," Management Science, vol. 15, pp. 215-227, 1969.

[4] F. Black, M. Scholes, "The pricing of options and corporate liabilities," Journal of Political Economy, vol. 81(3), pp. 637–654, 1973.

[5] J. Cox, S. Ross, A. Rubinstein, "Option pricing: a simplified approach," Journal of Financial Economics, vol. 7(3), pp. 229-263, 1979.

[6] A. Damodaran, "The promise and perils of real options," SSRN Working Papers, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1295849, 2006.

[7] E. Dziawgo, „Opcje rzeczywiste w zarządzaniu wartością przedsiębiorstwa," in Zarządzanie wartością firmy w dobie kryzysu, Kasiewicz S., Pawłowicz L., CeDeWu, Warszawa, pp 178–188, 2003.

[8] P. Fernandez, "Valuation and value creation in Internet-related companies," SSRN Working Papers, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=265609, 2001.

[9] P. Fernandez, "Valuing real options: frequently made errors," SSRN Working Papers, http://ssrn.com/abstract=274855, 2001.

[10] L. Klein, "Creating virtual product experiences: the role of telepresence," Journal of Interactive Marketing, vol.17(1), pp. 41-55, 2003.

[11] P. Kossecki, „Wartość życiowa klientów i jej zastosowanie do wyceny przedsiębiorstw," Problemy zarządzania, vol. 17(3), pp. 78-88, 2007.

[12] P. Kossecki, „Wycena i budowanie wartości przedsiębiorstw internetowych," Wydawnictwa Akademickie i Profesjonalne, Wyższa Szkoła Przedsiębiorczości i Zarządzania im. L. Koźmińskiego, Warszawa, 2008.

[13] A. La Plante, "Second life opens for business," Information Week, http://www.informationweek.com/story/showArticle.jhtml?articleID=197008342, 2007.

[14] J. Mizerka, „Opcje rzeczywiste w finansowej ocenie efektywności inwestycji," Wydawnictwo Uniwersytetu Ekonomicznego w Poznaniu, Poznań, 2005.

[15] A. Polis, „Koncepcje wyceny opcji realnych," in Opcje realne w przedsięwzięciach inwestycyjnych, W. Rogowski, Szkoła Główna Handlowa w Warszawie, Oficyna Wydawnicza, Warszawa, 2008.

[16] T. Stanisz, „Funkcje jednej zmiennej w badaniach ekonomicznych," PWN, Warszawa, 1986.

[17] P. Szczepankowski, „Wycena i zarządzanie wartością przedsiębiorstwa", PWN, Warszawa, 2007.

[18] U. Świerczyńska-Kaczor, „Wirtualne światy – nowe wyzwania dla menadżerów marketingu", Problemy zarządzania, vol. 23(4), pp. 179-196, 2008.

[19] U. Świerczyńska-Kaczor, Model budowy interakcji przedsiębiorstwa z użytkownikami wirtualnych światów, Problemy zarządzania, vol. 25(2), pp 185–202, 2009.

Advanced Stiff Systems Detection

Václav Šátek, Jiří Kunovský and Jan Kopřiva
University of Technology, Faculty of Information Technology,
Božetěchova 2, 61266 Brno, Czech Republic
Email: kunovsky@fit.vutbr.cz

***Abstract*—The paper deals with stiff systems of differential equations. To solve this sort of system numerically is a difficult task.**

Generally speaking, a stiff system contains several components, some of them are heavily suppressed while the rest remain almost unchanged. This feature forces the used method to choose an extremely small integration step and the progress of the computation may become very slow. However, we often need to find out the solution in a long range. It is clear that the mentioned facts are troublesome and ways to cope with such problems have to be devised.

There are many (implicit) methods for solving stiff systems of ordinary differential equations (ODE's), from the most simple such as implicit Euler method to more sophisticated (implicit Runge-Kutta methods) and finally the general linear methods. The mathematical formulation of the methods often looks clear, however the implicit nature of those methods implies several implementation problems. Usually a quite complicated auxiliary system of equations has to be solved in each step. These facts lead to immense amount of work to be done in each step of the computation.

On the other hand a very interesting and promising numerical method of solving systems of ordinary differential equations based on Taylor series has appeared. The question was how to harness the said "Modern Taylor Series Method" for solving of stiff systems. The potential of the Taylor series has been exposed by many practical experiments and a way of detection and solution of large systems of ordinary differential equations has been found.

These are the reasons why one has to think twice before using the stiff solver and to decide between the stiff and non-stiff solver.

***Index Terms*—Stiff systems, Taylor series terms, Modern Taylor Series Method, TKSL.**

I. Introduction

This paper is related with computer simulations of continuous systems. The research group HPC ("High performance computing") [8] has been working on extremely exact and fast solutions of homogenous differential equations, nonlinear ordinary and partial differential equations, stiff systems, large systems of algebraic equations, real time simulations and corresponding software and hardware (parallel) implementations since 1980.

The "Modern Taylor Series Method" (MTSM) is used for numerical solution of differential equations. The MTSM is based on a recurrent calculation of the Taylor series terms for each time interval. Thus the complicated calculation of higher order derivatives (much criticised in the literature) need not be performed but rather the value of each Taylor series term is numerically calculated.

An important part of the MTSM is an automatic integration order setting, i.e. using as many Taylor series terms as the defined accuracy requires. Thus it is usual that the computation uses different numbers of Taylor series terms for different steps of constant length.

The MTSM has been implemented in TKSL software [13]. Some articles that are focused on the MTSM were published last years [15], [16], [17], [18].

There are several papers that focus on computer implementations of the Taylor series method in different context "a variable order and variable step" (see, for instance, [1], [3]). Another more detailed description of a variable step size version and software implementation of the Taylor series method can be seen in [11]. The stability domain for several Taylor methods is presented in [2]. Promising A-stable combination of implicit Taylor series method with Trapezoidal rule is described in [9], [10].

A. Modern Taylor Series Method

The best-known and most accurate method of calculating a new value of a numerical solution of ordinary differential equation $y' = f(t, y), \quad y(0) = y_0$ is to construct the Taylor series [6].

Methods of different orders can be used in a computation. For instance the order method ($ORD = n$) means that when computing the new value y_{i+1} uses n Taylor series terms in the form

$$\begin{aligned} y_{i+1} &= y_i + h \cdot f(t_i, y_i) + \frac{h^2}{2!} \cdot f'(t_i, y_i) + \\ &\quad + \cdots + \frac{h^n}{n!} \cdot f^{(n-1)}(t_i, y_i)\,, \qquad (1) \\ y_{i+1} &= y_i + DY1_i + DY2_i + \cdots + DYn_i\,, \qquad (2) \end{aligned}$$

where h is integration step and DY are Taylor series terms.

Similarly we can construct implicit Taylor series method of order n in the form

$$\begin{aligned} y_{i+1} &= y_i + h \cdot f(t_{i+1}, y_{i+1}) - \frac{h^2}{2!} \cdot f'(t_{i+1}, y_{i+1}) - \\ &\quad - \cdots - \frac{(-h)^n}{n!} \cdot f^{(n-1)}(t_{i+1}, y_{i+1})\,, \qquad (3) \\ y_{i+1} &= y_i + DY1_{i+1} + DY2_{i+1} + \cdots + DYn_{i+1}\,. \qquad (4) \end{aligned}$$

II. Stiff systems

One of the most frequently mentioned definition of stiff systems is to use stiffness ratio r [14].

Let

$$\mathbf{y}' = \mathbf{f}(y,t)\,, \tag{5}$$

be a system of k ordinary differential equations. Let $\mathcal{J}$ be the Jacobian of the system (5) and λ_i the eigenvalues of $\mathcal{J}$. The eigenvalues λ_i are generally time dependent. Let the eigenvalues λ_i be arranged in the following way:

$$\mathrm{Re}|\lambda_{max}| \geq \mathrm{Re}|\lambda_i| \geq \mathrm{Re}|\lambda_{min}|\,, \quad i = 1 \cdots k-2\,, \tag{6}$$

then the *stiffness ratio* is in the form

$$r = \frac{\mathrm{Re}|\lambda_{max}|}{\mathrm{Re}|\lambda_{min}|}\,. \tag{7}$$

The stiffness ratio r is a coefficient that helps to decide whether a problem is stiff or not. A higher r indicates a more stiff system. However, there is no exact value of the stiffness ratio r that would distinguish the non-stiff problems from the stiff-problems. For many problems in common practice the stiffness ratio r is "very high" (say $1 \cdot 10^6$ or higher).

A. Test example 1

Let us examine "school example" system

$$\begin{aligned} y' &= -ay\,, \quad a > 0\,, \\ z' &= -0.0001z\,, \end{aligned} \tag{8}$$

with initial conditions $y(0) = 1$, $z(0) = 1$.

Note: well known analytic solution of (8) is in the form

$$\begin{aligned} y &= e^{-at}\,, \\ z &= e^{-0.0001t}\,, \end{aligned} \tag{9}$$

Typically we calculate the Jacobian of the system (8)

$$\mathcal{J} = \begin{pmatrix} -a & 0 \\ 0 & -0.0001 \end{pmatrix}\,,$$

then we specify the eigenvalues of the system (8)

$$\begin{aligned} \lambda_1 &= -a\,, \\ \lambda_2 &= -0.0001\,. \end{aligned}$$

We suppose that $a > 0.0001$ then the stiffness ratio of the system is in the form

$$r = \frac{\mathrm{Re}|\lambda_{max}|}{\mathrm{Re}|\lambda_{min}|} = \frac{a}{0.0001}\,. \tag{10}$$

Many stiff systems solver needs to compute the Jacobian of the ODEs systems to detect the stiffness. Modern Taylor Series Method as implemented in TKSL software needn't compute Jacobian matrix or eigenvalues of the ODEs systems.

Explicit Taylor series solution of (8) is in the form

$$\begin{aligned} y_{i+1} &= y_i - ah \cdot y_i + \frac{(-ah)^2}{2!} \cdot y_i + \\ &\quad + \cdots + \frac{(-ah)^n}{n!} \cdot y_i\,, \end{aligned} \tag{11}$$

$$y_{i+1} = y_i + DY1_i + DY2_i + \cdots + DYn_i\,, \tag{12}$$

similarly

$$\begin{aligned} z_{i+1} &= z_i - 0.0001h \cdot y_i + \frac{(-0.0001h)^2}{2!} \cdot z_i + \\ &\quad + \cdots + \frac{(-0.0001h)^n}{n!} \cdot z_i\,, \end{aligned} \tag{13}$$

$$z_{i+1} = z_i + DZ1_i + DZ2_i + \cdots + DZn_i\,. \tag{14}$$

Let us analyze Taylor series terms in the first step. Taylor series terms DZ have rapidly decreasing trend. As we can see in Fig. 1 for constant $a = 1$ (respectively $r = 10000$) and integration step size $h = 1$ we need 15 Taylor series terms to obtain result with absolute error $EPS = 10^{-10}$.

When the constant a is increased (the stiffness ratio r is also increased) we need to use more Taylor series terms to keep the stability of numerical method. In Fig. 2 resp. Fig. 3 we can see Taylor series terms for $a = 10$ ($r = 100000$) resp. $a = 100$ ($r = 1000000$). To obtain local error $EPS = 10^{-10}$ for $a = 100$ we need to use 295 Taylor series terms.

There is a problem when $a = 100$ (Fig. 3). As we can see in Fig. 3 Taylor series terms DY haven't got decreasing trend and we have to use multiple arithmetic.

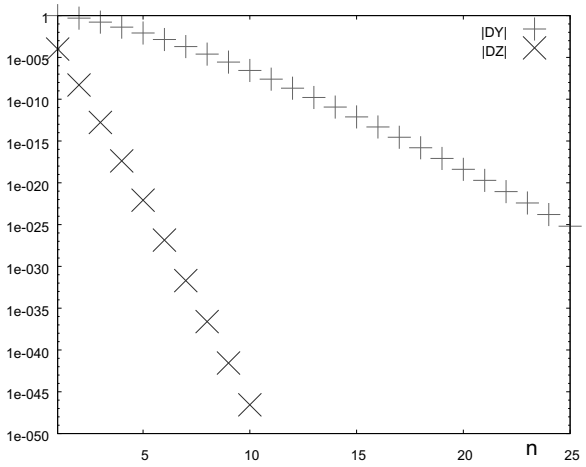


Fig. 1. Taylor series terms, $a = 1$

As we can see in Fig. 3 and Fig. 2 absolute value of Taylor series terms DY have increasing character. Modern Taylor series method as implemented in TKSL automatically detects (from different and rapidly growing Taylor series terms) the stiffness in system (8) with growing constant a and automatically reduces integration step size h. Tendency of decreasing Taylor series terms after automatic decreasing step size is shown in Fig. 4.

Conclusion: The TKSL automatically detects stiff system (8) using Taylor series terms and automatically reduces integration step size until the strategy in Fig. 4 is obtained. After detection of stiffness (using explicit MTSM), implicit Taylor series method must be used as presented in the Test example 2 as follows.

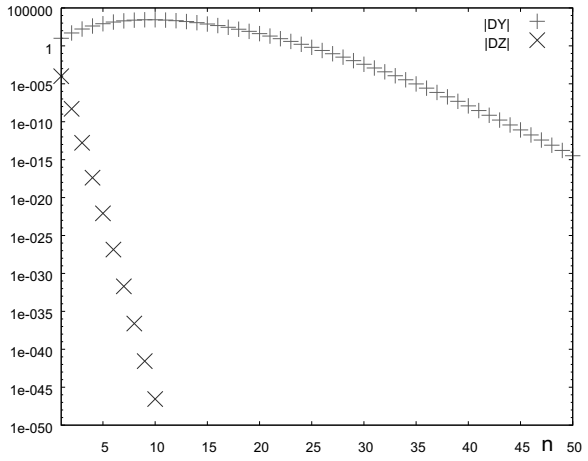


Fig. 2. Taylor series terms, $a = 10$

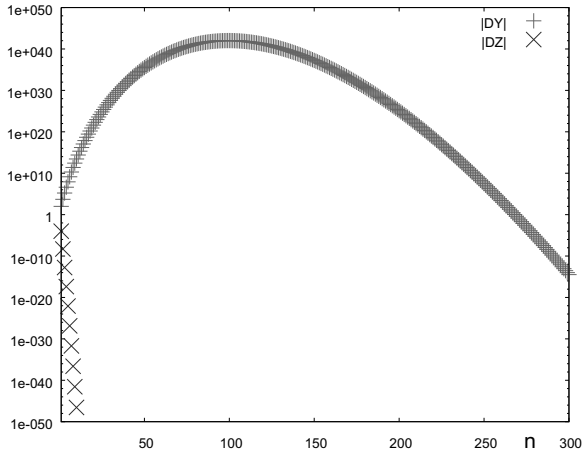


Fig. 3. Taylor series terms, $a = 100$

B. Test example 2

Stiff systems in some literature [7] are defined as systems of ODEs where explicit numerical methods don't work and implicit numerical methods must be used.

Let us analyze system [14]

$$\begin{aligned} y' &= z\,, \\ z' &= -b\cdot y-(b+1)\cdot z\,, \qquad b\in(1,\infty)\,, \end{aligned} \tag{15}$$

with initial conditions $y(0) = 1$, $z(0) = -1$.

Well known analytic solution of (15) is in the form

$$\begin{aligned} y &= e^{-t}\,, \\ z &= -e^{-t}\,. \end{aligned} \tag{16}$$

The system (15) becomes stiff for $b \gg 0$ and stiffness ratio is $r = b$.

Let's try to find the solution of (15) with explicit Taylor series method. Absolute error of numerical solution which is defined as difference between numerical z_i and analytical $z(t_i)$ solution

$$|\text{Error}(z)| = |z_i - z(t_i)|\,, \tag{17}$$

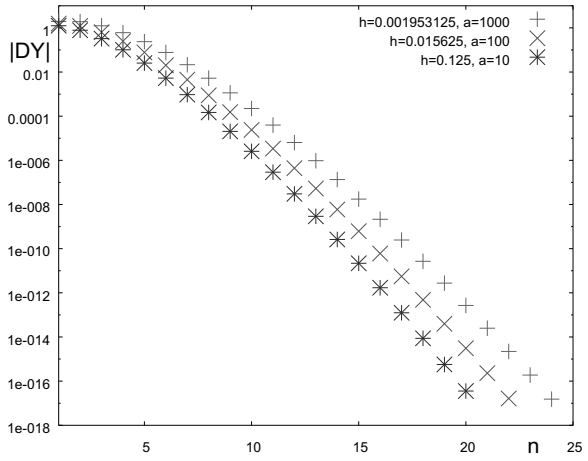


Fig. 4. Taylor series terms after automatic step size reduction

where $t_i = h \cdot i$ is shown in Tab. I. Abbreviation $ORD = 1$ means that 2 Taylor series terms are used during the computation (explicit Euler method) in Tab. I. explicit Euler method becomes unstable with growing constant a according to the Tab. I. We should reduce integration step size, or we must use more Taylor series terms Tab. II.

TABLE I
ABSOLUTE ERROR: EXPL. TAYLOR SERIES METHOD, $h = 0.1$, $ORD = 1$

	$\lvert\text{Error}(z)\rvert$			
t	$b = 10^4$	$b = 10^5$	$b = 10^6$	$b = 10^7$
0.1	0.00483742	0.00483742	0.00483742	0.00483742
0.2	0.00873075	0.00873075	0.00873075	0.00873075
0.3	0.0118182	0.0118182	0.011818	0.011781
0.4	0.01422	0.0143073	0.0375021	37.2672
0.5	0.0160768	0.856836	2328.16	3.72529×10^{7}
0.6	0.0187225	8727.91	2.32816×10^{8}	3.72529×10^{13}

TABLE II
ABSOLUTE ERROR: EXPL.TAYLOR SERIES METHOD, $h = 0.1$, $b = 100$

	$\lvert\text{Error}(y)\rvert$			
t	$ORD = 1$	$ORD = 2$	$ORD = 3$	$ORD = 4$
0.1	0.00483742	0.000162582	4.0847×10^{-6}	8.1964×10^{-7}
0.2	0.00873075	0.000294247	7.39197×10^{-6}	1.48328×10^{-7}
0.3	0.0118182	0.000399404	1.00328×10^{-5}	2.0132×10^{-7}
0.4	0.01422	0.000481905	1.2104×10^{-5}	2.4302×10^{-7}
0.5	0.0160407	0.000545106	1.36903×10^{-5}	3.14994×10^{-7}
0.6	0.0173706	0.000591932	1.48514×10^{-5}	1.20207×10^{-5}

Multiple arithmetic: with growing constant a multiple arithmetic is needed. Only 9 Taylor series terms are used for integration step $h = 0.1$ and local error $EPS = 10^{-20}$. Corresponding word length of Taylor series terms for large constant b are shown in Tab. III.

TKSL automatically detects stiffness in system (15) (when b is growing) from Taylor series terms and TKSL uses

TABLE III
MULTIPLE ARITHMETIC

b	Word length [bits]
10^{10}	3681
10^{20}	8900
10^{50}	24000

automatically smaller step size.

Implicit Taylor series method (as implemented in iTKSL software) has prosperous properties to solve stiff systems especially implicit Taylor series method has bigger absolute stability domain than those of explicit Taylor series method.

Let's solve the system (15) with implicit Taylor series method. Implicit Taylor series is in the form

$$\begin{aligned} y_{i+1} &= y_i + hy'_{i+1} - \cdots - \tfrac{(-h)^n}{n!} y^{(n)}_{i+1}\,, \\ z_{i+1} &= z_i + hz'_{i+1} - \cdots - \tfrac{(-h)^n}{n!} z^{(n)}_{i+1}\,, \end{aligned} \tag{18}$$

where higher derivations are in the form

$$\begin{aligned} y'_{i+1} &= z_{i+1}\,, \\ z'_{i+1} &= -by_{i+1} - (b+1)z_{i+1}\,, \\ \\ y''_{i+1} &= -by_{i+1} - (b+1)z_{i+1}\,, \\ z''_{i+1} &= -by'_{i+1} - (b+1)z'_{i+1} = -bz_{i+1} - \\ &\quad -(b+1)(-by_{i+1} - (b+1)z_{i+1})\,, \\ \\ y'''_{i+1} &= -bz_{i+1} - (b+1)(-by_{i+1} - (b+1)z_{i+1})\,, \\ z'''_{i+1} &= -b(-by_{i+1} - (b+1)z_{i+1}) - (b+1)\cdot \\ &\quad \cdot(-bz_{i+1} - (b+1)(-by_{i+1} - (b+1)z_{i+1}))\,, \\ \\ y^{(4)}_{i+1} &= -b(-by_{i+1} - (b+1)z_{i+1}) - \\ &\quad -(b+1)(-bz_{i+1} - (b+1)(-by_{i+1} - (b+1)z_{i+1}))\,, \\ z^{(4)}_{i+1} &= -b(-bz_{i+1} - (b+1)z'_{i+1}) - \\ &\quad -(b+1)(-bz'_{i+1} - (b+1)(-bz_{i+1} - (b+1)z'_{i+1}))\,, \\ &\vdots \end{aligned}$$

After substitution higher derivations into implicit Taylor series form (18) we obtain numerical solution in the form

$$\begin{aligned} y_{i+1} &= \tfrac{y_i(hb+h+1)+z_i(h)}{1+h^2b+hb+h}\,, \\ z_{i+1} &= -\tfrac{y_i(hb)-z_i}{1+h^2b+hb+h}\,, \end{aligned}$$

for implicit Taylor series order 1 ($ORD = 1$) that is implicit Euler method.

Similarly for $ORD = 2$ we obtain formula in the form

$$\begin{aligned} y_{i+1} &= 2\tfrac{y_i(h^2b^2+h^2b+2hb+h^2+2h+2)+z_i(h^2b+h^2+2h)}{2h^2+2h^3b^2+h^4b^2+2h^3b+4+4hb+4h+4h^2b+2h^2b^2}\,, \\ z_{i+1} &= -2\tfrac{y_i(h^2b^2+2hb+h^2b)+z_i(h^2b-2)}{2h^2+2h^3b^2+h^4b^2+2h^3b+4+4hb+4h+4h^2b+2h^2b^2}\,, \end{aligned}$$

implicit Taylor series $ORD = 3$ is in the form

$$\begin{aligned} y_{i+1} &= 6(y_i(h^3b^3 + h^3b^2 + h^3b + h^3 + 3h^2b^2 + 3h^2b + 3h^2 + \\ &\quad 6hb + 6h + 6) + z_i(h^3b^2 + h^3b + h^3 + 3h^2b + 3h^2 + \\ &\quad +6h))/(36 + 36h^2b + 6h^3b^3 + 36hb + 3h^5a^3 + \\ &\quad +3h^5b^2 + h^6b^3 + 18h^2 + 18h^3b^2 + 36h + 18h^3b + \\ &\quad +9h^4a^2 + 6h^4b^3 + 6h^4b + 6h^3 + 18h^2b^2)\,, \\ z_{i+1} &= -6(y_i(3h^2b^2 + 6ha + 3h^2b + h^3b + h^3b^2 + h^3b^3) + \\ &\quad +z_i(-6 + 3h^2b + h^3a^2 + h^3b))/(36 + 36h^2b + \\ &\quad +6h^3b^3 + 36hb + 3h^5b^3 + 3h^5a^2 + h^6b^3 + \\ &\quad +18h^2 + 18h^3b^2 + 36h + 18h^3b + 9h^4b^2 + \\ &\quad +6h^4b^3 + 6h^4b + 6h^3 + 18h^2b^2)\,, \end{aligned}$$

etc.

Absolute error of numerical solution using implicit Taylor series method of different order is shown in Tab. IV. Note that constant b has no influence on error of computation.

TABLE IV
ABSOLUTE ERROR: IMPLICIT TAYLOR SERIES METHOD, $h = 0.1$

	|Error(y)|: $b = 10^4, 10^5, 10^6, 10^7, 10^8$			
t	$ORD = 1$	$ORD = 2$	$ORD = 3$	$ORD = 4$
0.1	0.00425349	0.000139958	3.48077×10^{-6}	6.93811×10^{-8}
0.2	0.00771553	0.000253297	6.29908×10^{-6}	1.25557×10^{-7}
0.3	0.0104966	0.000343816	8.54948×10^{-6}	1.70413×10^{-7}
0.4	0.0126934	0.000414829	1.03145×10^{-5}	2.05595×10^{-7}
0.5	0.0143907	0.000469227	1.16662×10^{-5}	2.32538×10^{-7}
0.6	0.0156623	0.000509528	1.26673×10^{-5}	2.52491×10^{-7}

There is a problem with general formulation of y_{i+1}, z_{i+1} from implicit Taylor series formula (18) - other implicit numerical methods have the same problem. We have to use some iteration method to compute y_{i+1}, z_{i+1} in implicit form.

Implicit Taylor series method with recurrent calculation of Taylor series terms and Newton iteration (ITMRN) was implemented. Absolute errors of ITMRN of $ORD = 1, 2, 3, 4$ are the same as explicit calculations presented in Tab. IV and only two Newton iterations are needed. Absolute errors of ITMRN $ORD = 5, 6, 7$ and number of Newton iterations j which are needed to obtain numerical results with tolerance $TOL = 10^{-10}$ are shown in Tab. V.

TABLE V
ABSOLUTE ERROR: ITMRN $h = 0.1$, $ORD = 5, 6, 7$, $TOL = 10^{-10}$, $b = 10^4$

	$ORD = 5$		$ORD = 6$		$ORD = 7$	
t	|Error(y)|	j	|Error(y)|	j	|Error(y)|	j
0.1	1.153×10^{-9}	3	1.644×10^{-11}	3	2.035×10^{-13}	5
0.2	2.087×10^{-9}	3	2.976×10^{-11}	3	3.459×10^{-13}	5
0.3	2.833×10^{-9}	3	4.04×10^{-11}	3	4.842×10^{-13}	5
0.4	3.418×10^{-9}	3	4.874×10^{-11}	3	5.898×10^{-13}	5
0.5	3.866×10^{-9}	3	5.513×10^{-11}	3	6.354×10^{-13}	4
0.6	4.198×10^{-9}	3	5.986×10^{-11}	3	6.998×10^{-13}	5

Problem with increasing the number of Newton iterations j with increasing order of implicit Taylor series method is

shown in Tab. V. We should use multiple arithmetic with growing constant a and order of ITMRN or we should reduce integration step size to obtain better stability of ITMRN (Tab. VI). Absolute errors of ITMRN $ORD = 8$ and number of Newton iterations using in each step are shown in Tab. VI. There are two integration step size used - $h_1 = 0.1$ with absolute error $|\text{Error}_1(y)|$ and number of Newton iterations j_1 resp. $h_2 = 0.05$ with absolute error $|\text{Error}_2(y)|$ and number of Newton iterations j_2. The same arithmetic (double precision) is used in both cases.

TABLE VI
ABSOLUTE ERROR: ITMRN $h_1 = 0.1, h_2 = 0.05, ORD = 8$, $TOL = 10^{-10}, b = 10^4$

t	$\|\text{Error}_1(y)\|$	j_1	$\|\text{Error}_2(y)\|$	j_2
0.05	—	—	7.10543×10^{-15}	4
0.1	2.148×10^{-12}	9	1.77636×10^{-15}	4
0.15	—	—	5.00711×10^{-14}	4
0.2	2.753×10^{-14}	8	9.17044×10^{-14}	4
0.25	—	—	8.71525×10^{-14}	5
0.3	6.328×10^{-14}	19	8.23785×10^{-14}	5
⋮	⋮	⋮	⋮	⋮
0.6	4.938×10^{-12}	7	1.77636×10^{-14}	4

III. CONCLUSION

A very interesting and promising numerical method of solving systems of ordinary differential equations based on Taylor series has appeared. The question was how to harness the said "Modern Taylor Series Method" for solving of stiff systems. The potential of the Taylor series has been exposed by many practical experiments and a way of detection and solution of large systems of ordinary differential equations has been found.

Detailed information will be given during the INFORMATICS 2011 conference.

ACKNOWLEDGMENT

The paper includes the solution results of the Ministry of Education, Youth and Sport research project No. MSM 0021630528. This paper has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic.

REFERENCES

[1] R. Barrio, F. Blesa, and M. Lara. High-precision numerical solution of ODE with high-order Taylor methods in parallel. *Monografías de la Real Academia de Ciencias de Zaragoza*, 22:67–74, 2003.

[2] R. Barrio. Performance of the Taylor series method for ODEs/DAEs. *Applied Mathematics and Computation*, 163:525–545, 2005. ISSN 00963003.

[3] R. Barrio, F. Blesa, and M. Lara. VSVO Formulation of the Taylor Method for the Numerical Solution of ODEs. *Computers and Mathematics with Applications*, 50:93–111, 2005.

[4] D. Barton. On Taylor Series and Stiff Equations. *ACM Transactions on Mathematical Software*, 6(3), 1980.

[5] A. Gibbons. A Program for the Automatic Integration of Differential Equations using the Method of Taylor Series. *The Computer Journal*, 3:108–111, 1960.

[6] E. Hairer, S. P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I*, vol. Nonstiff Problems. Springer-Verlag Berlin Heidelberg, 1987, ISBN 3-540-56670-8.

[7] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, second revised ed. with 137 Figures, vol. Stiff and Differential-Algebraic Problems. Springer-Verlag Berlin Heidelberg, 2002, ISBN 3-540-60452-9.

[8] *High Performance Computing*, web pages, http://www.fit.vutbr.cz/~kunovsky/TKSL/index.html.en.

[9] X. Chang, Y. Wang, and L. Hu. An implicit Taylor series numerical calculation method for power system transient simulation. In *MIC'06 Proceedings of the 25th IASTED international conference*, pages 82–85, 2006.

[10] X. Chang, H. Zheng, and X. Gu. An A-stable improved Taylor series method for power system dynamic-stability simulation. In *Proceedings of the Power and Energy Engineering Conference Asia-Pacific*, pages 1–5, 2010.

[11] A. Jorba and M. Zou. A software package for the numerical integration of ODE by means of high-order Taylor methods. *Exp. Math.*, 14(1):99–117, 2005.

[12] G. Kirlinger and G. F. Corliss. On implicit taylor series methods for stiff odes. In *The Proceedings of SCAN 91: Symposium on Computer Arithmetic and Scientific Computing*, pages 371–379, 1991.

[13] J. Kunovský, *Modern Taylor Series Method*, Habilitation work, FEI-VUT Brno, 1994.

[14] E. Vitásek, *Základy teorie numerických metod pro řešení diferenciálních rovnic*, Academia, Praha, 1997, ISBN 80-200-0281-2.

[15] J. Kunovský, M. Pindryč, V. Šátek, F. V. Zbořil. Stiff Systems in Theory and Practice. In: *Proceedings of the 6th EUROSIM Congress on Modelling and Simulation*, Vde, AT, 2007, p. 6, ISBN 978-3-901608-32-2.

[16] J. Kunovský, V. Šátek, M. Kraus, J. Kopřiva. Automatic Method Order Settings. In *Proceedings of Eleventh International Conference on Computer Modelling and Simulation EUROSIM/UKSim2009*. Cambridge, GB, IEEE CS, 2009, p. 117–122, ISBN 978-0-7695-3593-7.

[17] J. Kunovský, M. Kraus, V. Šátek, V. Kaluža. New Trends in Taylor Series Based Computations. In *Proceedings of 7th International Conference of Numerical Analysis and Applied Mathematics*. Rethymno, Crete, GR, AIP, 2009, p. 282–285, ISBN 978-0-7354-0705-3.

[18] J. Kunovský, P. Sehnalová, V. Šátek. Explicit and Implicit Taylor Series Based Computations. In *8th International Conference of Numerical Analysis and Applied Mathematics*. American Institute of Physics, Tripolis, GR, 2010, s. 587–590, ISBN 978-0-7354-0831-9.

[19] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. SIAM, Philadelphia, 2009. ISBN 978-0-898716-69-6.

Agents in Multicore Realm

Márk Török, Zalán Szűgyi, Norbert Pataki
Department of Programming Languages and Compilers,
Eötvös Loránd University
Pázmány Péter sétány 1/C H-1117 Budapest, Hungary
Email: patakino@elte.hu

***Abstract*—Multi-agent models (i.e. more agents in the models) play an important role in simulations. Its main feature is decentralization: the agents do not connect to a central controlling unit. Each agent possesses limited information. The pieces of information about the environment are decentralized. During the simulation, there can be hundred, thousand or million agents alive and communicate with each other. The communication among agents is scalable. The interactions among them decide the structure of the model. GRID Systems or GPUS are widely-used because the efficiency and speedup belong to the present questions of agent-based simulations.**

Our aim is to support multicore-based agents. We implement existing models for multicore environment with the help of Cilk++ that is one of the most well-known languages extension of C++ providing new keywords for multicore programming. Our framework supports generation of Cilk++ code, too.

I. Introduction

Agent-based modeling is a powerful simulation modeling technique that has seen a number of applications in the last few years, including applications to real-world business problems. In agent-based modeling (ABM), a system is modeled as a collection of autonomous decision-making entities called agents. Each agent individually assesses its situation and makes decisions on the basis of a set of rules. Agents can include firms, people within organizations, or entire industries: Agent-based modeling allows the user to define the interactions among agents and then use these to generate models.

Agent-based models are particularly suitable for modeling complex systems, where many agents interact; there is therefore scope for using these toolkits to illustrate some of the more recent concepts within management, particularly those where the micro level behavior of individuals affects the global, macro properties of a system [1].

Multicore programming is an interesting new way of programming. Nowadays multicore processors are in use in many kind of applications [2]. However, agent-based models are typically use grids instead of one processor. We intend to use agent-based models in a multicore environment. Cilk++ is an ideal multicore programming language to support agents. Cilk++ code can be generated from Fables models.

This paper is organized as follows. We introduce the Fables language in section II briefly. The Cilk++ with our extension is shown in III. Our approach present in section IV, and finally this paper concludes in section V.

II. Fables

The basic concept of agent-based modeling is to create adaptive agents to operate in a changing environment. Agents make autonomous decisions and modify their environment through continuous interactions. The *Functional Agent-Based Language for Simulations* (FABLES) is a special purpose language for ABM that is intended to reduce programming skills required to create simulations [3]. The aim of FABLES is to allow modelers to focus on modeling, and not on programming.

The language was designed to have no more language elements than required for a common model. It has defined structures for each specific part of the simulation: for the model and the agents (object-oriented approach), for the behaviour (functional approach) and for the model dynamics (initialization, stopping, agent interactions scheduling). In this way the Fables source code is very compact and straightforward in general [4].

Let us consider the following FABLES code snippet:

```
model Ants {
  param defaultSeed = 12345;
  param antNum      = 100;
  worldSize         = 100;
  norm (x)          = x mod worldSize;

  class Ant
  {
    var pos;
    move (x) = pos := norm( pos + x );
    schedule Stepper cyclic 1
    {
      1 : move(
        discreteUniform(-1, 0, 1) );
    }
  }; // class{ Ant }

  world = [ a.pos : a is Ant ];
  startUp( defaultSeed, antNum )
  {
    seed( defaultSeed );
    [ create Ant[ pos := worldSize div 2 ] :
        i is [1..antNum] ];
  }
}; // model{ Ants }
```

The model has to have some input parameters, as the seed or the number of the ants thats living in the colony. We have to define the world size that the ants live in. Fables is a functional language, so we easely define a function, call it norm with one parameter. This method guarantees that no ant will go over the border of our worlds. The representation of an ant is a class, that has a field, called `pos`. This value contains the current position of an ant in the world. The method `move` has a parameter and after a step it sets the new position up. The norm method, that we define above called here. Every entity or class in Fables has a stepper method. This method is being called during the simulation. Every calling means a change in the world in the simulation and changing of the given entity. This method is automaticly called by the controller. Finally, we initialize the world and start up the simulation with the ants.

Java code can be generated from FABLES models [5]. However, now we can generate Cilk++ code, too.

III. CILK++

The recent trend to increase core count in processors has led to a renewed interest in the design of both methodologies and mechanisms for the effective parallel programming of shared memory computer architectures. Those methodologies are largely based on traditional approaches of parallel computing.

The Cilk++ language can be used to efficiently execute our application on multicore machines [6]. In this language, applications run in the Cilk++ runtime, which manages parallel execution using computation workers. These workers run on separate Operating System threads and there is one worker per CPU core. The Cilk++ language is C++ with some additional language features. Parallel work is created when the keyword `cilk_spawn` precedes the invocation of a function. The semantics of spawning differ from a C++ function (or method) call only in that the parent can continue to execute in parallel with the child, instead of waiting for the child to complete as is done in C++. The scheduler in the Cilk++ runtime system takes the responsibility of scheduling the spawned functions on the individual processor cores of the multicore computer. A function cannot safely use the values returned by its children until it executes a `cilk_sync` statement. The cilk sync statement is a local "barrier", not a global one as, for instance, is used in message-passing programming. In addition to explicit synchronization provided by the `cilk_sync` statement, every Cilk function syncs implicitly before it returns, thus ensuring that all of its children terminate before it does. Loops can be parallelized by simply replacing the keyword for with the keyword `cilk_for` keyword, which allows all iterations of the loop to operate in parallel [7]. Cilk++ is a C++ extension, thus C++ Standard Template Library can be used as general framework for containers and algorithms. On the other hand, STL is not optimized for multicore environment [8]. In this way, STL can be an efficiency bottleneck as it does not support multicore programming.

We have developed a multicore implementation of C++ STL for the Cilk++ platform [9]. We have reimplemented the containers, as well as, the STL's generic algorithm that are container-independent. There are several operations on vectors, which can be improved by parallelism [10].

IV. APPROACHES

In this section we present our approach, hence we show the generated code. This code is similar to C++ or Java, but it takes advantage of multicore architecture We have generated Cilk++ code from the FABLE model.

First, the FABLE model's `Ant` class is translated to a Cilk++ class. the constructors, member functions and the data members come from the model:

```
// ant.h:

#ifndef ANT__H_FN4ARI3G
#define ANT__H_FN4ARI3G

class Ant
{
public:
  Ant( int p ) : pos( p )
  {
  }

  void move( int x )
  {
    pos = norm( pos + x );
  }

private:
  int pos;
};

#endif
```

Second, we generate a compilation unit and a header file for function `norm`:

```
// norm.h:
int norm( int );

// norm.cpp:
int norm( int x, int worldSize )
{
  return x % worldSize;
}
```

Our framework supports some predefined functions for simulation, for example `discreteUniform`:

```
template <class T>
const T& discreteUniform( const T* p,
                          int N )
{
  return p[ rand() % N ];
}
```

And finally, we create a compilation unit for the entry point of the simulation. This part of code uses our multicore `vector` implementation and it takes advantage of the Cilk++'s `cilk_for` loop, too.

```
// main.cpp:

#include "functions.h"
#include "norm.h"
#include "ant.h"
#include <cstdlib>

int main()
{
  const int defaultSeed = 12345;
  const int antNum = 100;
  const int worldSize = 1000;
  const int v[] = { 1, 0, -1 };
  // startUp
  srand( defaultSeed );
  vector<Ant> world;
  cilk_for( int i = 0; i < antNum; ++i )
  {
    world.push_back(
      Ant( worldSize / 2 )
    );
  }

  while( true )
  {
    cilk_for( int i = 0;
              i < world.size();
              ++i )
    {
      world[i].move(
       discreteUniform(
         v,
         sizeof( v ) / sizeof( v[0] )
      );
    }
  }

}
```

This is the Cilk++ code that our framework generates from the FABLE model. Moreover the framework is highly customizable on several purposes such as: output or lifecycle.

V. Conclusion

Multicore programming has become one of the most important platform. However, multicore programming has many successful application areas, but agent-based modeling hardly uses this new platform for simulations.

In this paper we have investigated the prospects of a multicore multi-agent models. We use the widely-used FABLES language to model and our framework generates multicore Cilk++ code from these models. The generated code as a proof of our concept also uses our multicore Cilk++ STL library.

References

[1] L. Gulyás, T. Kozsik, and J. B. Corliss, "The Multi-Agent Modelling Language and the Model Design Interface," *Journal of Artificial Societies and Social Simulation*, vol. 2, no. 3, October 31 1999, dBLP Record 'journals/jasss/GulyasKC99'. [Online]. Available: http://www.soc.surrey.ac.uk/JASSS/2/3/8.html

[2] H. Bischof, S. Gorlatch, and R. Leshchinskiy, "Generic parallel programming using c++ templates and skeletons," in *Domain-Specific Program Generation*, ser. Lecture Notes in Computer Science, C. Lengauer, D. Batory, C. Consel, and M. Odersky, Eds. Springer Berlin / Heidelberg, 2004, vol. 3016, pp. 107–126.

[3] R. Legéndi, L. Gulyás, R. Bocsi, and T. Máhr, "Modeling autonomous adaptive agents with functional language for simulations," in *Progress in Artificial Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, vol. 5816, pp. 449–460.

[4] L. Gulyás, S. Bartha, T. Kozsik, R. Szalai, A. Korompai, and G. Tatai, "The Multi-Agent Simulation Suite (MASS) and the Functional Agent-Based Language of Simulation (FABLES)," 2005, extended abstract for SwarmFest 2005, Turin, Italy. [Online]. Available: http://www.swarm.org/wiki/SwarmFest2005/Program

[5] M. Iványi, R. Bocsi, L. Gulyás, V. Kozma, and R. Legéndi, "The multi-agent simulation suite," in *Emergent Agents and Socialities: Social and Organizational Aspects of Intelligence*, 2007.

[6] C. E. Leiserson, "The cilk++ concurrency platform," in *Proceedings of the 46th Annual Design Automation Conference*, ser. DAC '09. New York, NY, USA: ACM, 2009, pp. 522–527.

[7] M. Frigo, P. Halpern, C. E. Leiserson, and S. Lewin-Berlin, "Reducers and other cilk++ hyperobjects," in *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures*, ser. SPAA '09. New York, NY, USA: ACM, 2009, pp. 79–90.

[8] S. Meyers, *Effective STL – 50 Specific Ways to Improve Your Use of the Standard Template Library*. Reading, MA: Addison-Wesley, 2001.

[9] Z. Szűgyi, M. Török, and N. Pataki, "Towards a multicore C++ Standard Template Library," in *Proc. of Workshop on Generative Technologies 2011 (WGT 2011)*, 2011, pp. 38–48.

[10] A. Alexandrescu, *Modern C++ Design*. Reading, MA: Addison-Wesley, 2001.

An Application of Business Process Modeling System ILNET

Dimitar Blagoev, George Totkov, Elena Somova
Department of Computer Science
The University of Plovdiv "Paisii Hilendarski"
Plovdiv, Bulgaria
gefrix@gmail.com, totkov@uni-plovdiv.bg,
eledel@uni-plovdiv.bg

Abstract — **In this paper we present a system for accompanying of the process of Bachelor and Master thesis administration, including all stages: publication of thesis themes, thesis choice, documents fulfillment, thesis writing, thesis submission, thesis review, thesis publication, etc. The system is developed on the base of business process modeling system ILNET and its basic model for describing and executing workflow models. The basic model allows the creation and usage of both high-level constructs that are available in all popular business modeling standards as well as low-level programming language constructs that allow the implementations of new building blocks. Amongst the main features of the system are the capability of the blocks to implement their own custom visualization using the same workflow model, the centralized independent hosting and execution of the models, the ability to hot-swap a compiled and running model with another (newer) one and the automatic wrapping of web services and .NET libraries into ILNET building blocks.**

Keywords – workflow, business process modeling, administration of education

I. INTRODUCTION

In this paper we introduce a novel process modeling environment and present an approach towards the automation of administrative processes. Still big portion of the administrative work at learning institutions is being carried by out manually on paper. Some of the processes are short and can be done fast, while other spread over a long period of time, involves collecting and processing data from multiple members and is more difficult to monitor and administer. The notion of modeling and executing business processes using workflows or other techniques has been around for decades. Initially in the late 1970s it was focused on office automation systems (Officetalk-Zero [2]), shortly after systems supporting organizational processes emerged (SCOOP [6], which used Petri Nets). Since then many new tools and approaches have been introduced and new standards emerged trying to help different tools exchange process definitions. Using Petri Nets and in particular Colored Petri Nets is a good choice in cases of computer algorithm analysis and improvement [4]. Presenting a good base for simulation of process models is probably one of the reasons some business process modeling systems chose CPN as their underlying model representation language, while some of the others focus on model visualization and visual description, not execution or analysis.

Amongst the vast number of tools and models two standards have become widely accepted – the BPMN (Business process modeling notation) and EPC (Event-driven process chain). These two seemingly completely different process modeling approaches are used by most of the recently developed tools in the field. The execution of the BPMN models has been possible with the help of the BPEL (Business process execution language) standard, while XPDL (XML process definition language) focuses more on the visual representation of the models (also known as Workflow process description language in its first revision from 1998, before it began supporting the BPMN constructs). The latter also works as an exchange format for process definitions between different systems (including those based on the EPC approach). The second version of the BPMN standard (version 2) also supports process execution.

In 1999 begins the "Workflow Patterns" initiative under the supervision of Prof. Wil van der Aalst. The goal of the initiative is to present a thorough study of the different perspectives (workflow control, data, resources and exception handling) that should to be supported by a process management or business process modeling standard. Currently there are 43 patterns for workflow control, 43 for working with resources, 40 for data and 108 (derived) for exception handling. Table 1 shows a comparison of some of the most widely used standards for process modeling (BPMN, UML, EPC) and process execution and management (BPEL, XPDL).

TABLE 1. SUPPORTED WORKFLOW PATTERNS BY SOME OF THE BUSINESS PROCESS MODELLING STANDARDS

Patterns	BPEL	BPMN	XPDL	UML	EPC
Workflow Control	21	33	33	30	12
Data	19	22	16	18	-
Resources	28	8	-	8	-
Exception Handling	16	32	18	-	-

Often new patterns are being added for either further simplifying the process modeling process or fixing problematic zones that arise from the use of already existing prior patterns. One issue which can be observed is that there isn't a single

standard that covers or can promise to cover all of the possible necessary patterns for optimal and seamless process description. Bellow we present a possible solution, by proposing a system in which new building elements and patterns can be defined in the same manner as processes.

II. SYSTEM ILNET

The ILNET system is a light framework for creating business process modeling solutions. It is the final product of a long and extended development of series of tools for linguistic processing. One of its main features is that, although a fixed base model with which processes can be defined and executed is given, there is a built-in feature to enhance and adapt both the graphical editor and the underlying engine, to define and execute processes using models with which the end users may be more familiar with. Currently Petri Net modeling is in testing and most of the BPML elements are drafted.

The first version of the system was designed to be used primarily in the field of computational linguistics [1]. The system is used successfully in the text-to-speech solution SLOG [5].

In [3] the new and current, now based on.NET, version of the system, called ILNET, is used during the creation of a prototype e-learning architecture for modeling e-learning processes related to the course workflow.

The system is modular and consists of five main components, which communicate between them through inside messages or SOAP-based web services:

- Workflow compiler, which translates process descriptions to programming code;
- Server for execution and management of the described processes;
- Graphical editor for visual process modeling;
- Event manager which handles internal communication between processes and provides a gateway for the outside environment to exchange messages with running process instances using SOAP;
- Building blocks library which contains descriptions of utility sub-processes which can be used for more rapid process modeling and also provides the ability to use familiar structures from other modeling tools.

Figure 1 presents the architecture of the system in common.

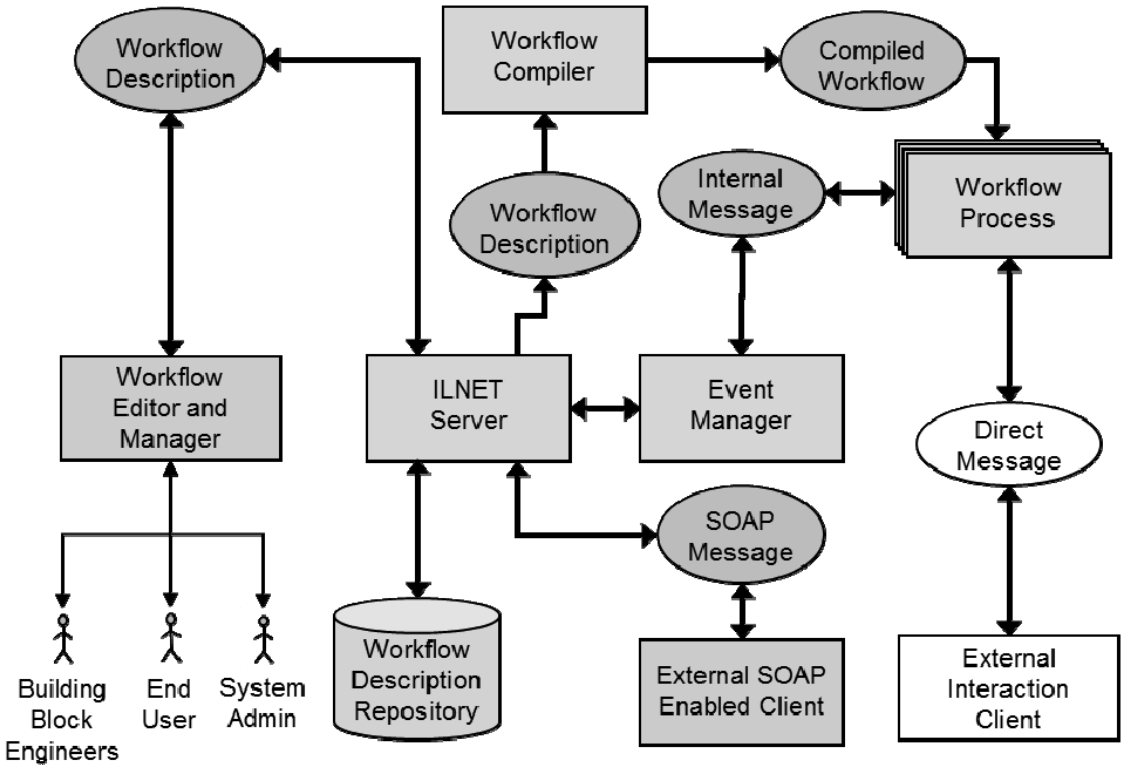


Figure 1. Arhitecture of ILNET

Another feature of the proposed system is its memory model. The language for process description in ILNET supports working with variables on three different access levels:

- Local variables: they are always copied and are accessed by only one instance. They can represent the memory of workflow thread, or Petri Net token. They are used for calculations which do not depend on the result of other simultaneously running process threads, eliminating the risk of accessing one variable being accessed from more than one place at the same time.
- Intermediate variables: they are copied when the process is initializing. They are accessed by each control workflow, derived from the initial (when the process is started). They are used for synchronization of the active roads in the frame of one calling of the process and for implementation of most of the templates used for modeling of business processes which deal with synchronization of workflows.
- Global variables: they are never copied and are available to all instances. These variables provide a way of creating process instance data pools where different instances of the same process collect and share data. For short-term data storage of non-critical data these variables are faster and easier to work with than using an external database (which is preferred for critical or long-term data).

Using all three types of variable access levels it is possible to easily model complex data workflows.

Loading and execution of the modeled processes and transferring of inner- and outer- process messages is realized by ILNET server. The server operates with events for work with messages and serves for connection between the surrounding environment and the running process instances giving opportunity for:

- loading process models;
- starting processes;
- receiving and sending messages;
- termination of processes instances.

The graphical editor for visual process modeling provides means for seamless process modeling in the ILNET framework. Figure 2 shows the working screen of the graphical interface of the module for visual process editing GEFI. The environment enables several processes to work simultaneously, they are synchronized in real time with the server, in which they are loaded and gives basic means for tracking and monitoring of executed instances of modeled processes.

The environment supports the use of different modeling standards using custom building blocks. The resulting visual information generated for a particular model is attached to the base elements as metadata.

One novel approach used in the tool is the use of a hybrid client-server rendering engine. While most systems provide either full-client or full-server rendering (for web clients) GEFI uses client-side rendering for the model whilst allowing server-side rendering for specific elements (building blocks) of the model. In essence, this means that the tool can be used to edit how the blocks are rendered inside the tool. In addition, since the rendering of the block is handled by the block description on the side of the server, it is possible to model complex interactive multi-user elements.

III. Application System

The paper presents an application of the business process modeling system ILNET and its basic model for describing and executing workflow models, introduced in the previous section. The application system has a purpose to follow and maintain the whole administrative process of Bachelor and Master thesis preparation of the students for the graduation of the respective university degree.

The process covers the following stages: publication of the thesis themes, thesis choice, administrative acceptance, documents fulfillment, thesis writing with support with help materials (as thesis templates and information about university thesis rules and time schedule about each stage of preparation of the graduation thesis), thesis submission and publication, additional materials (as thesis defense presentation, code of the developed software, software documentation, published scientific papers and other prepared materials) publication, thesis review, thesis evaluation and thesis view .

Administrative acceptance is done on two stages of the process, according to the university rules – first, tutor has to accept the concrete student for the chosen theme and second, the head of the department has to accept the student to work on thesis with the respective tutor (*Application Form 1*). During the whole process two official paper documents (application forms) have to be fulfilled. The second official document – *Application Form 2* is a request for permission of defending the thesis in front of the evaluation commission.

The user types that can use the opportunity of the system are student, teacher-tutor, teacher-evaluator and teacher-head.

The tutor publicizes the thesis themes that he/she gives, views and updates own themes, views received requests from students to work on a particular theme, answers to the students' requests (accepts or rejects) and reviews the thesis. The tutor can follow all steps of the student during the whole process of preparation of the thesis.

The head, which is the head of the department, approves the application form 1 and determines the reviewer of the thesis.

The evaluator, which is a member of the thesis evaluation commission, has the opportunity to see the list of themes for defense and publishes the final thesis mark.

The student chooses the thesis theme from the list of free themes, fills in two application forms (which is almost automatic), views informational and help materials, and submits thesis and additional thesis materials.

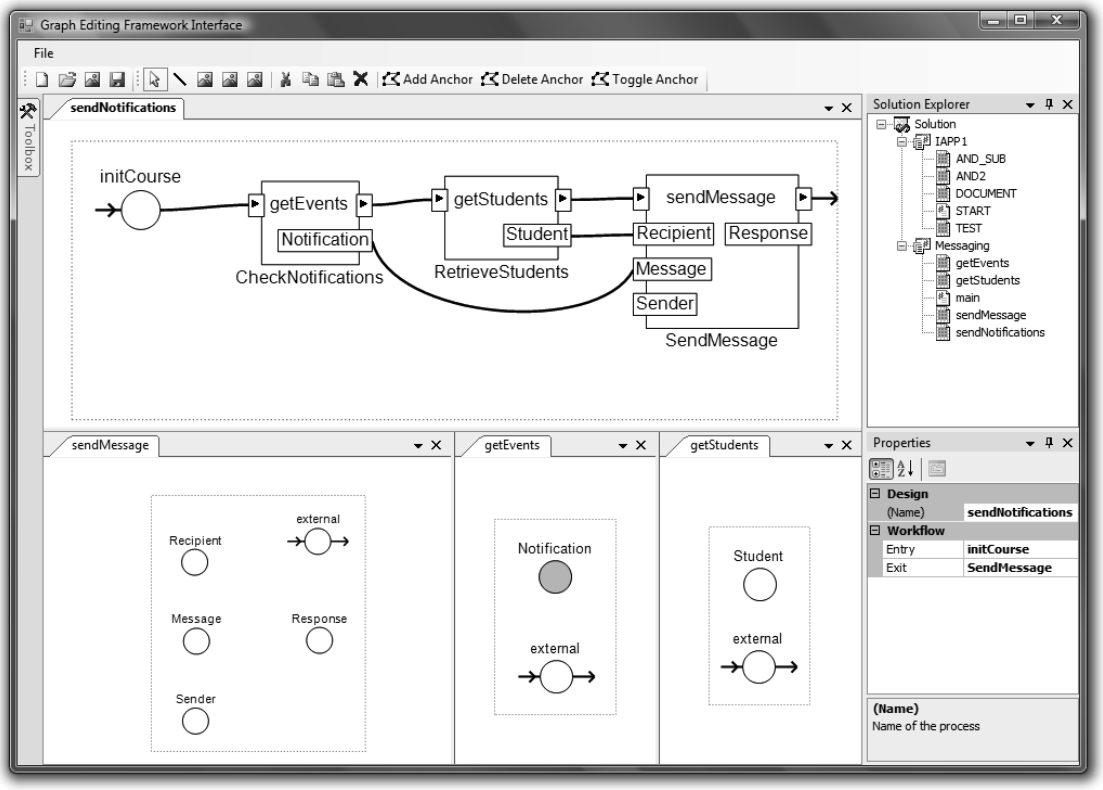


Figure 2. Grafical environment for process edition

Different views are prepared for each type of user – free themes, own themes, themes for defense, *Application Form 1* requests, current thesis with status, review requests and finished thesis.

The chosen approach about passing over processes for all users is to use one workflow for all users instead of different workflows for each of them. The used approach gives easy management of processes and control of the relationship between processes executed by different users.

The scheme of the thesis administrative process is given on the Figure 3. On the figure the possible actors of the actions are pointed with *s*, *t*, *e* and *h* respectively for student, tutor, evaluator and head.

Passing over the workflow is consecutively with some time interruptions when particular user has to wait other user to execute some action. For example, while tutor do not accept student request for specific theme, student cannot continue the process and start to prepare the thesis.

In the process there are some stages that depend by the time limitation – fulfillment of both application forms and all final submission (thesis, materials, review and evaluation mark). Passing through these stages can be done by user choice or reaching the definite time.

In the workflow there is used one special type of ILNET blocks, which presents *and*-logic in execution. For example, only after when the thesis, the additional materials, the review and the *Application Form 2* (according to the university rules) are submitted, the thesis work is prepared and ready for evaluation.

The application system is developed on the base of the following software technologies: MS Visual Studio 2010, MS Silverlight 4.0 and LINQ and database MS SQL Server.

The work is funded by projects DO 02-308 to the National Science Fund and BG051 PO 001-3.3.04/13 to European Social Fund of Operational Program “Development of the human resources” 2007-2013.

IV. Conclusion

The work presents example how to realize an application system on the base of business process modeling system using workflow models. The system follows the whole administrative process of preparation of Bachelor and Master graduation thesis. We intend to expand the system with connection to the university student system for automatic decision if the student has the right to prepare thesis according to the evaluation marks.

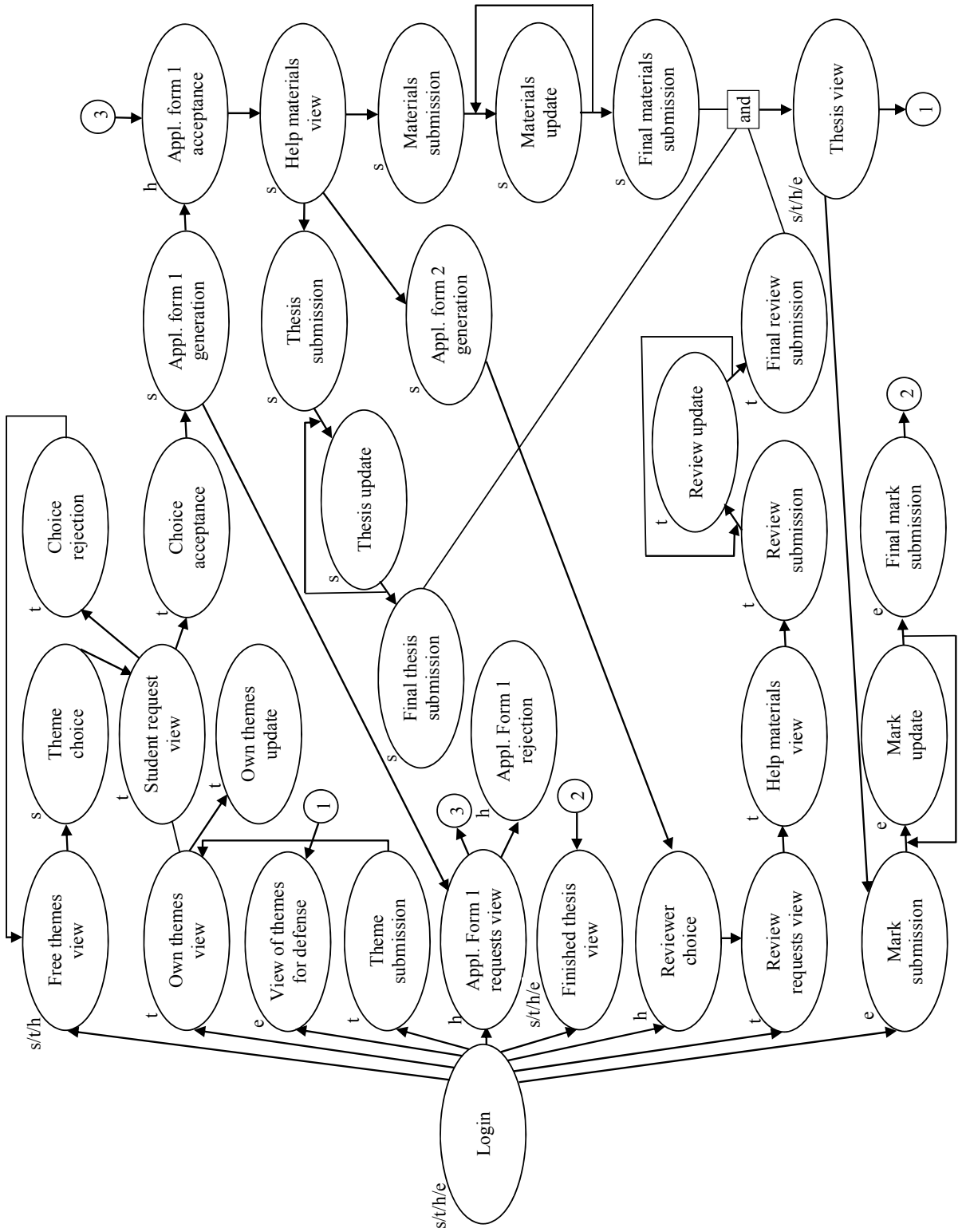


Figure 3. Workflow model of the application system

REFERENCES

[1] Blagoev D., G. Totkov, Visual Parser Builder, RANLP'05, Borovetz, 112-116.

[2] Clarence M., A. Ellis, G. J. Nutt, Office Information Systems and Computer Science. *ACM Comput. Surv.* 12, 1 (March 1980), 27-60. DOI=10.1145/356802.356805, http://doi.acm.org/10.1145/356802.356805

[3] Indzhov Hr., D. Blagoev, G. Totkov, Executable Petri Nets: Towards Modelling and Management of e-Learning Processes, ACM International Conference Proceeding Series; Vol. 375, Proc. of the 10th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing 2009, Rousse, Bulgaria, June 18-19, 2009. IIIA.12.1 IIIA.12.6.

[4] Korec□ko S., B. Sobota, C. Szabó, Performance analysis of processes by automated simulation of Coloured Petri nets, 10th International Conference on Intelligent Systems Design and Applications (ISDA), 2010, Cairo, Nov. 29 2010-Dec. 1 2010, p176-181.

[5] Totkov G., D. Blagoev, V. Angelova, SLOG 1.0: Bulgarian text to speech transformation system, International Joint Conf. on Computer, Information, Systems Sciences and Engineering (CIS2E'05), Bridgeport (CT), Dec 10-20, 2005.

[6] Zisman M., Representation, Specification and Automation of Office Procedures. PhD Dissertation. Department of Business Administration, Wharton School, University of Pennsylvania, Philadelphia, PA, 1977.

Automation of experimenting with new various input data traffic models

J. Tot'
Department of Computer Science and Engineering
University of West Bohemia
Pilsen, Czech Republic
jtot@kiv.zcu.cz

P. Herout
Department of Computer Science and Engineering
University of West Bohemia
Pilsen, Czech Republic
herout@kiv.zcu.cz

***Abstract*—Traffic control is one of the current key issues to increase road safety and to minimize congestion and pollution in urban areas. To achieve optimized traffic control, new traffic models are developed featuring algorithms profiting of different input data for particular city areas. Current advanced traffic simulating systems and modelling tools are based on various traffic models, but they have limited possibilities for modification or replacement of the in-built traffic models and corresponding data inputs. Development of new traffic models encompasses repetitive composition of experiments from different data inputs. It is difficult to create an experiment if larger amount of data has to be input manually into the model. Our object is to create an open environment in which traffic models could be connected to various data inputs and experiments with different settings could be launched and evaluated. To simplify the development process, a system prototype based on Extensible Stylesheet Language (XSL) configurations is proposed to automate the compositions of experiments. The prototype is tested on a particular traffic control model developed at the Institution of Information Theory and Automation of the Academy of Sciences of the Czech Republic. The prototype allows for testing the model on larger areas (North of Zličín in Prague, Czech Republic) and due to the automation, the duration of the experiment compositions was reduced to seconds.**

Keywords-traffic modelling; road traffic control; traffic simulation system; experiments

I. INTRODUCTION

Computer simulation is a method, which, due to the increasing computational performance, has influenced many areas. The increasing number of vehicles, and the associated traffic problems, gave rise to research in road traffic simulation in order to improve on traffic control. One of the current key issues is to increase safety on roads, to minimize congestions and, especially in urban areas, to minimize the negative impact of emissions and noise produced in road transport, see [1]. Hence many different traffic models were developed and many others are still being developed, especially to cover particularities dominant in complex city areas.

There are many computerized analytical tools – advanced traffic modelling tools (ATMT). They integrate transport modelling software for wide professional applications common in commercial sphere [2]. These tools are very complex and sophisticated systems and they provide user-friendly environment created exactly to fit requirements of the in-built modelling algorithms making them relatively easy to reapply the simulations with different input settings for various situations. Such tools are e.g. Aimsun [4], [5], Paramics [4], Vissim [4], JUTS [6], etc.

In a case, when a new model is developed, one of the above mentioned or similar tools can be used. As illustrated in Fig. 1, such a tool consists of two parts. One part represents one or more "in-built general models" – core modelling algorithms (CMA) – and the second part stands for all services of environment and utilities (SEU). The SEU allows for processing input data set (IDS) within the CMA and it provides user-friendly support to the developer (user/experimenter) during the developing/modelling process. Generally, the development process encompasses network modelling (e.g. network analysis and particular network elements definition and characteristic), traffic modelling (e.g. traffic flows and control plan programming) and model adjusting (calibration and verification), see [1]. ATMT usually provide automations of most of the parts of the development process and thus it is possible to use them to model large simulation areas in a relatively short time [3].

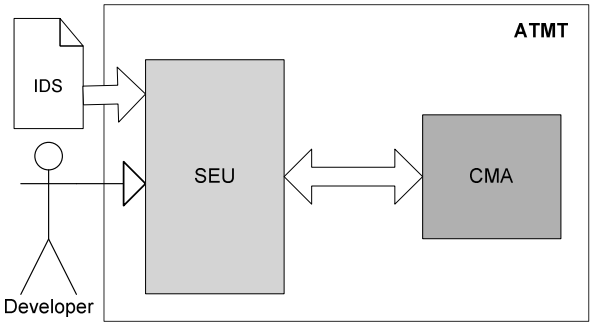


Figure 1. Schema of a general ATMT consisting or CMA and SEU, where SEU stands in-between the CMA, IDS and the developer.

In general, ATMT with in-built CMA and rich SEU guarantee rapid model development, open architecture, API and other interfaces for connection with additional systems. Nevertheless, these tools are limited in following cases:

- The experimental model development requires modification of CMA. (ATMT may have various options, but they are in-built and cannot be modified in any other way than they permit in advance.)
- Modelling process consists of a comparative study among various CMA. (The in-built CMA cannot be replaced.)
- Data required for the modelling are different – IDS origins from various resources, its incomplete or in diverse formats. (ATMT require a predefined IDS in particular format suitable for the CMA.)

It's obvious that developing new traffic models that would not be restricted in above mentioned cases cannot be performed in conventional ATMT. This means that the user-friendly services of SEU cannot be used and the development process requires additional effort to either create new SEU that could cooperate with the CMA (very complex and difficult task, see [5], [6]), or pass through the development process without support services of the environment.

According to [7], the computational complexity of simulation model calibration is NP-complete. When a traffic model development is in the adjusting phase, verification and calibration are required in order to tune the model parameters. The more complex the model is, the more experiments – which run simulations with different IDS, settings or even different CMA – are required.

II. MODEL DEVELOPMENT EXPERIMENTS

Let's say that the first experiment E_1, performed during the development process of the model, corresponds to an experiment composition (including simulation run) r_1 in order to IDS I_1 is processed within the CMA A_1 to get results R_1. An experiment E_1 can be expressed as

$$\mathrm{E}_1\colon \{\mathrm{I}_1, \mathrm{A}_1\} \xrightarrow{r_1} R_1. \tag{1}$$

Developer makes decision D_1 within the process d_1 on the basis of result R_1 .about settings of second experiment E_2, which is needed to be performed with a new set of data IDS I_2 (modification of I_1) within the CMA A_2 (modification of A_1). The decision D_1 is

$$\mathrm{D}_1\colon R_1 \xrightarrow{d_1} \{\mathrm{I}_2, \mathrm{A}_2\} \tag{2}$$

and the second experiment E_2 based on the result of decision D_1 is

$$\mathrm{E}_2\colon \{\mathrm{I}_2, \mathrm{A}_2\} \xrightarrow{r_2} R_2, \tag{3}$$

where r_2 and R_2 are analogically the second simulation run, respectively result of second experiment.

If the development process is simplified to a performance of *n* experiments, it is possible to claim, analogically to previous statements (1), (2) and (3), that the development process *P* can be expressed as a series

$$P\colon \{\mathrm{I}_1, \mathrm{A}_1\} \xrightarrow{r_1} R_1 \xrightarrow{d_1} \{\mathrm{I}_2, \mathrm{A}_2\} \xrightarrow{r_2} R_2 \dots\ R_{n-1} \xrightarrow{d_{n-1}} \{\mathrm{I}_n, \mathrm{A}_n\} \xrightarrow{r_n} R_n. \tag{4}$$

The part $\{\mathrm{I}_\mathrm{n}, \mathrm{A}_\mathrm{n}\}$ presents the final settings of the model, and R_n is the result the corresponding experiment provides. In another words, the development process can be understand as a series of experiments compositions and developer's decisions

$$P'\colon \{\mathrm{r}_1, \mathrm{d}_1, \mathrm{r}_2, \mathrm{d}_2, \dots, \mathrm{r}_\mathrm{n}, \mathrm{d}_\mathrm{n}\}, \tag{5}$$

where the series

$$R'\colon \{\mathrm{r}_1, \mathrm{r}_2, \dots, \mathrm{r}_\mathrm{n}\} \tag{6}$$

represents all the experiment compositions (including all the simulation runs) and the series

$$D'\colon \{\mathrm{d}_1, \mathrm{d}_2, \dots, \mathrm{d}_\mathrm{n}\} \tag{7}$$

represents all the model developer's decision making processes.

If the IDS $I_1, I_2, \dots I_n$ of the traffic model are entered manually to the corresponding CMA $A_1, A_2, \dots A_n$ and there are no utilities to process the results, the processes in series (5) make the development of the model extremely time-consuming and error-prone, see [10]. The series (4) and (5) indicate that the higher number of experiments *n* represents, the more significant the issue is. The number *n* grows with the complexity of the model so the manual development of the model becomes very limited and it doesn't allow for developing models manually with larger IDS.

III. OPEN TRAFFIC SIMULATION ENVIRONMENT APPROACH

Our proposal is an approach based on an open traffic simulation environment (OTSE) that serves to develop new traffic models that cannot be raised in the current ATMT. Furthermore, as well as the ATMT, this environment represents a system, that attempts to simplify the development process and to avoid the problems of time-consuming and error-prone repetitive manual actions. Developer's intervention shouldn't be required during process r_i to get result R_i from IDS I_i and CMA A_i of any experiment E_i where

$$\mathrm{E}_i\colon \{\mathrm{I}_i, \mathrm{A}_i\} \xrightarrow{r_i} R_i. \tag{8}$$

This could be achieved with the corresponding set of SEU that permits to automatize the actions r_i and to help the user in decision making process (7) and consequently development of models with larger IDS can be possible.

Based on these ideas, an OTSE system is proposed in Fig. 2 (The Fig. 2 is an extension of the Fig. 1). The system provides similar SEU allowing for IDS to be processed within CMA and to support the model developer during the developing/modelling process. Besides, communication interfaces I1, I2 and I3 between IDS and SEU, and SEU and CMA in both directions, were proposed in the way as it is demonstrated in Fig. 2. The structures of data that are transferred via these interfaces could be configured arbitrarily with corresponding

configurations C1, C2 and C3 so that various IDS/CMA could be used within the system. This also means that an adequate C1 is necessary to provide with each different IDS and adequate C2 and C3 are necessary to provide with each different CMA.

In comparison with conventional ATMT, this configurations-based OTSE enables removing, upgrading and swapping both IDS and CMA. Because of this, the limits of conventional ATMT could be crossed. OTSE permits: model development requiring modification of CMA; modelling using results from different algorithms; using data including inputs from various resources, incomplete data sets or inputs in diverse formats.

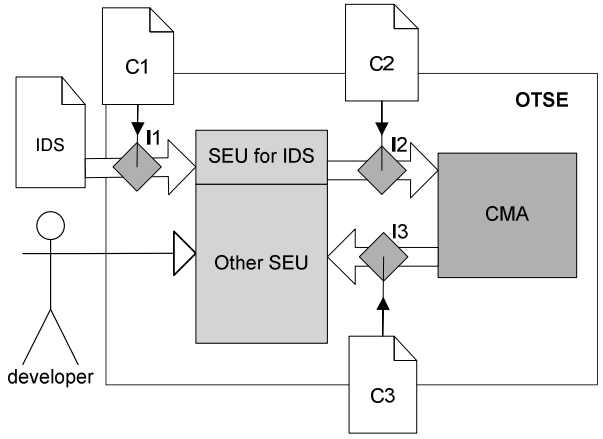


Figure 2. Schema of proposed configurations-based OTSE approach.

A. Prototype implementation

The proposed OTSE was implemented as a standalone Java desktop application built on XML and XML-related technologies. The configurations C1, C2 and C3 are sets of Extensible Stylesheet Language (XSL) files. The prototype integrates an open source XSL Transformation (XSLT) processor, which is able to process the XSL files from each configuration. High scalability of XSL allows for writing transformative sheets for wide range of input/output data structures. The system accepts any IDS or CMA for which the XSL configurations 1, 2 and 3 are prepared – and these are limited only by the XSLT technology. For XSLT, the requirements include: First, the IDS must be available in open file formats. Second, the CMA must feature inputs reading and results writing similarly in/to any open file format. Third, all data files should be easily parsed and processed within the XSLT processor (e.g. XML, TXT or CSV files). The Fig. 3 outlines and example how a configuration could be used in order to transform files of different formats from IDS with XSL files from C1 within the XSLT processor (which is part of SEU) to XML documents holding the transformed data in inner OTSE data structures and formats (files A.XML, B.XML, C.XML,… which are generally different from files from IDS). The example in Fig. 3 emphasizes first transformation controlled by 1.XSL in which 1.TXT is transformed to A.XML within the XSLT processor. The example indicates also associations among the XSL files and the source files from IDS, which are considered as inputs for each transformation controlled via the corresponding XSL document from the configuration.

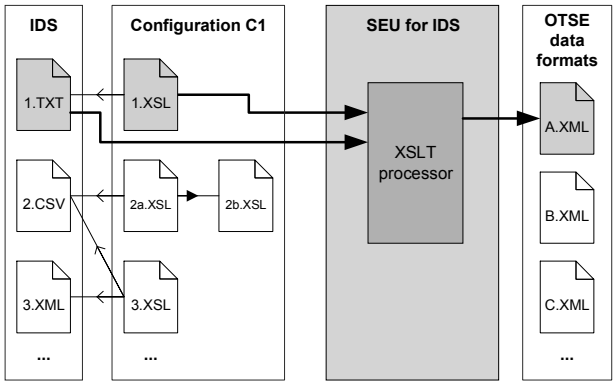


Figure 3. Example of C1 and its role among IDS, XSLT processor and inner OTSE data formats.

There are some SEU with user-friendly GUI available to support the developer's decision making processes (7) and XSLT utilities provided for automation of the composition of experiments in order to make each r_i in (8) run without user intervention. Experiment compositions are automatic one-click set of XSLT transformations, as it is introduced in [10], [11]. The most important SEU utilities implemented in the prototype are following:

1) Maps Manager

Maps Manager is a tab panel that allows multiple simulation areas to be contained within a single application running instance, which helps the decision process it encompasses experiments on different simulation areas comparison.

2) Map Viewer

Map Viewer (MV) is a canvas for rendering the selected data of the map inside the application. It is implemented with the Apache Batik SVG Toolkit and it is able to visualize the simulation area exactly as it was designed in the originating editor.

3) Properties Viewer

Properties Viewer is a tree-view panel with a user filter showing selected elements of the loaded inputs in DOM.

4) Properties Editor

Properties Editor is a utility that allows editing any data from IDS in special user-friendly editors if the input data types are specified.

5) CMA integrator

CMA integrator is a utility supporting simulation run processes (6) allowing for one click automation of experiment composing and launching based on configurations 2 and 3 (see Fig. 2).

6) Experiments Manager

Experiments Manager is an explorer and manager that serves to manage all available experiments (and their results) performed and automatically stored in application, see [12]. It is designed to support the developer's decision making processes (7).

7) Experiments Results Viewer

Experiments Results Viewer is an evaluating tool used to visualize the results of experiments and to support developer's decision making processes (7). Its functionality is described in [12].

8) TediTransformer

TediTransformer (TT) is a particular utility dedicated to automatic transformations. It transforms maps saved in Aimsun Traffic Editor (TEDI) ASCII formats to XML structure used by the prototype application. Its functionality is described in [10].

9) Connections to another systems

Utilities MV and TT were designed to cooperate together to be able to import, view and edit maps from the TEDI. This demonstrates that the prototype of OTSE can cooperate easily with external utilities, even with the editor from conventional ATMT. Because of this, our prototype takes advantages of the TEDI from Aimsun, see [10], [11]. The traffic model development, which would have not been performed in this type of tool due to the limits listed in *Introduction*, may be used if combine with our prototype. In this case, any simulation area can be created in TEDI, then imported to our prototype application in TT and after that viewed in MV and partially edited and used for experimenting with selected CMA together with other data from IDS.

There is a screenshot of GUI of main window together with some SEU of the prototype desktop application in Fig. 4. There are MV (with visible sample intersection), Properties Viewer (panel on the right), Experiments Results Viewer (dialog window with the figure), Experiments Manager (dialog window at the bottom) and bar at the top providing access to CMA integrator, TT and another SEU.

IV. CASE STUDY AND RESULTS

The proposed prototype implementation was tested with one of the specialized traffic control simulation model (test model), which is being developed at the Institution of Information Theory and Automation of the Academy of Sciences of the Czech Republic. In [8], it is explained that the objective of the test model is to simulate the effects of various control plan settings on a selected urban road area. Conventional ATMT cannot be used to support the experimenting with the test model because of the following reasons: Firstly, the development of the model requires modification of CMA, secondly, the data designated for the modelling origin from various resources and are in diverse formats, and thirdly, the model is designated for comparative study among various CMA, see [8], [9]. On this account, it is assumed that this model is fully suitable to be tested as a case study of our prototype.

A. IDS

The IDS required by the model encompasses files from various resources. Besides the common inputs – simulation area topology, traffic demand, control plans and simulation global settings – the IDS of the model contains also the data obtained from real traffic data measurements as evolutions of occupancy and intensity collected on road detectors within selected time periods. The formats used in IDS are TXT, CSV, XML document and ASCII Aimsun map files, see [10]. All of these formats are open, so they fulfil the prototype implementation requirements.

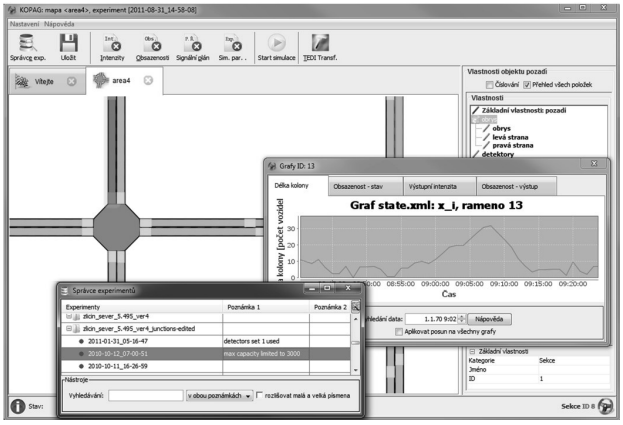


Figure 4. Screenshot of GUI of SEU of the prototype desktop application with some of the implemented SEU.

B. CMA

The CMA of the model feature special methods that are experimentally developed to utilize all available information, especially real traffic data measurements. This information is then used to estimate resulting queues lengths on arms of selected intersections after simulations with different traffic control plans are performed. According to [11], the CMA are available in two versions – as a Matlab application and Java application – which are easily configurable for the OTSE.

C. Testing and results

There were three tests performed with the test model in our prototype of OTSE:

1) Test 1

The objective of the first test was to determine if OTSE is applicable and if it allows for experimenting with the test model. In this test, the test model was used with an area of simple intersection of three arms. This area corresponds to a real intersection in Prague (GPS location is 50° 3' 22.800" N, 14° 17' 13.350" E) and it is visualized in Fig. 5. If the OTSE is not used, it takes from minutes to hours – depending on the changes performed in comparison to previous experiments – to compose a single new experiment manually. First the XSL configurations 1, 2 and 3, according to the IDS and requirements of CMA, were prepared. Then the selected area of intersection map was designed in TEDI. After that, the map was imported using TT into our prototype of OTSE. The result is positive –it is possible to perform the development process (editing, launching, evaluating and managing the simulations experiments in the) using the SEU in our prototype. The left part of the Fig. 5 shows the used simulation area loaded in the prototype application.

2) Test 2

The objective of this test was to measure minimal experiment composition duration r_i and the duration that a developer spends with a simplest experiments performed in OTSE. The test model with the same simulation area as in the test 1 was used (see the map on the left of Fig. 5). The prototype was

launched on an average PC (Intel Core2 Duo CPU P8600, 2.96 GB RAM, Windows 7 x86). Two durations were measured for a set of following synthetic experiments – a set of 23 experiments in that one input parameter was being changed (the speed limit of one road of the test intersection was being increased from 20 kmph to 130 kmph with a step of 5 kmph). Each experiment was edited and launched one by one (moreover, each experiment is automatically saved in order to be later available in Experiments Manager). The decision making process was intentionally skipped (no check of the results of experiments during the test) so that the duration of (7) is zero. Only Properties Editor was used to edit the data before launching each experiment with CMA Integrator. It was measured that during this test, the average duration of composition of such experiment using the selected SEU was 13 sec. (with editing one input parameter) and the duration of r_i (without editing) was 4 sec., see summarization in Table I.

3) Test 3

The objective of the last test was to repeat the test 2 on a larger area. For that reason a network of intersections in North part of Zličín in Prague was selected. This area encompasses also the intersection used in the tests 1 and 2 – it is the northernmost intersection in the simulation area – see the imported map on the right of Fig. 5. It was measured that during this test, the average duration of composition of such experiment using the selected SEU was 14 sec. (with editing one input parameter) and the duration of r_i (without editing) was 5 sec., see summarization in Table I.

TABLE I. EXPERIMENTS COMPOSITION DURATION WITH THE TEST MODEL USING OTSE

Testing area	Average composition duration r_i	
	Including simple editing	***Excluding editing***
One intersection[a]	13 sec.	4 sec.
Network of intersections[a]	14 sec.	5 sec.

a. North part of Zličín in Prague, Czech Republic

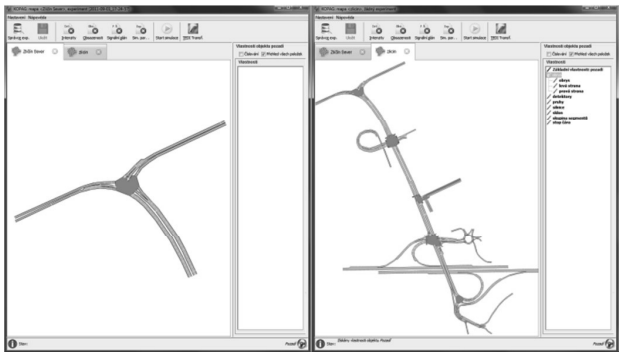


Figure 5. Visualisation of test simulation areas – a simple intersection (on the left) and a larger area (on the right) – both from the part of Northern Zličín in Prague, Czech Republic.

V. CONCLUSION

In this paper, a complex and sophisticated open traffic simulation environment was proposed. It is suitable for such situations when conventional advanced traffic modelling tools cannot be used, but supportive services and utilities are essential for successful model development. This system is premised on configurations-based architecture that allows for that any core modelling algorithms and their corresponding data input set can be used via the supportive services and utilities. The presented approach was realized as a fully working prototype application, which implements the proposed system in Java and XSL. It features basic services/utilities in order to automate the processing of data inputs, to reduce significantly the amount of information entered manually, and to simplify as much as possible the repetitive work of an experimenter. The prototype was tested with a real traffic control model and promising results were obtained. The prototype makes possible rapid compositions of experiments (duration of single experiment composition was 4-5 seconds) comparable with the standard of conventional modelling tools. This allows for using the test model on larger simulation areas than before.

REFERENCES

[1] J. Martínez, V. R. Tomás, J. Samper, J. J. Martínez, I. Sáchez, F. Soriano, "Simulación de tráfico. Systemas de control de tráfico. (Traffic simulation. Traffic control systems.)", Universidad de Valencia, 2009.

[2] Federal Highway Administration – United States Department of Transportation. Traffic Analysis Tools by Category. [Online]. 2004. Available: http://ops.fhwa.dot.gov/trafficanalysistools/tat_vol2/sectapp_e.htm

[3] F. García, C. Estefanía, J. M. Aguirre, J. M. Larrauri, V. R. Tomás, J. Samper, "El uso de simuladores como herramienta de apoyo en la gestión del tráfico (The usage of simulators as a supportive tool for traffic control)," Departamento de Obras Públicas y Transportes, Universidad de Valencia, 2009.

[4] P. Hidas, "A functional evaluation of the AIMSUN, PARAMICS and VISSIM microsimulation models," Road and Transport Research, vol. 14, pp. 45-59, Dec. 2005.

[5] J. Barceló, E. Codina, J. Casas, J. L. Ferrer and D. García, "Microscopic traffic simulation: A tool for the design, analysis and evaluation of intelligent transport systems," Journal of Intelligent & Robotic Systems, vol. 41, pp. 173–203, 2004.

[6] D. Hartman, P. Herout, "JUTS – J-Sim Urban Traffic Simulator," in Proceedings of the International Workshop Modelling and Simulation in Management, Informatics and Control (MOSMIC), Žilina, 2003.

[7] M. Hofmann, "On the Complexity of Parameter Calibration in Simulation Models," The Journal of Defense Modelling and Simulation Applications, Methodology, Technology, vol. 2 no. 4, pp. 217-226 Oct. 2005.

[8] J. Kratochvílová, I. Nagy, "Model dopravní mikrooblasti (Traffic model of a microregion)," ÚTIA AV ČR and Fakulta dopravní, Prague, 2004.

[9] I. Nagy, J. Homolová, P. Pecherková, "Dopravně závislé řízení silničního provozu ve městech," Prague, May 2007.

[10] J. Toť, "Application for data preparation of traffic simulation experiments," M.S. thesis, University of West Bohemia, Pilsen, Mar. 2010.

[11] P. Havránek, "Konverzní programy a konstrukce simulačního modelu dopravy (Converter programs and traffic simulation model construction)," M.S. thesis, University of West Bohemia, Pilsen, Mar. 2010.

[12] P. Kopač, "Rozhraní simulačního systému pro dopravní simulace (Simulation System User Interface for Traffic Simulation)," Bc. thesis, University of West Bohemia, Pilsen, Mar. 2010.

On the Usage of ELLAM to Solve Advection-diffusion Equation Describing the Pollutant Transport in Planetary Boundary Layer

Radim Dvorak, Frantisek Zboril
Faculty of Information Technology
Brno University of Technology
Bozetechova 2, Brno, Czech Republic
idvorak@fit.vutbr.cz, zboril@fit.vutbr.cz

Abstract— **The paper deals with the numerical solution of the advection diffusion equation describing the pollutant transport in the atmosphere. The idea is to use the not so common ELLAM framework and to compare the results with the state of the art Walcek method, which is used to solve the advection problems. The real wind model was chosen for the tests in order to get the good performance idea of the two methods. From the performed experiments and the calculation times, one can conclude that ELLAM is suitable to solve the presented problem.**

Keywords-component; ELLAM, advection-diffusion equation, numerical solution, advection models

I. INTRODUCTION (HEADING 1)

The air pollution modelling is an important and very current topic. It allows predicting the progress of the pollutant dispersion from the specific time to the near future and thus to help to deal with the low-quality air or to prevent from the possible contamination.

The model of the pollutant transport is described by the specific partial differential equation (1). It consists of the several parts that describe the whole process. The first and the most important part from the pollutant behaviour point of view is the advection term. It determines the wind field, commonly changing during time and space, which is the most influencing term in the equation. The diffusion/dispersion is the second important part and also it changes during time and space. The rest of the equation can be summarized into the reaction term which includes the behaviour of the reactant in the atmosphere. All three mentioned parts are in balance with the other side of the equation. It consists of source term – the source(s) of the pollution.

$$\frac{\partial c}{\partial t} = \nabla \cdot (c\vec{u}) + \nabla \cdot \left(D \nabla c\right) + R(c) \qquad (1)$$

Variable c is the concentration, $\vec{u}$ is the velocity field, D is the diffusion-dispersion model, R is the reaction model and t is time.

The model is usually solved by operator splitting technique, where the equation is separated into the advection, diffusion and reaction models. All the models are then evaluated separately and the calculated concentration is counted together. Although the special simple methods can be used to solve the separate parts, the error due to splitting, which is often neglected, arises [1].

One of such a method was proposed by [2] and it was successfully tested and compared against up to its date best methods that solve the pure advection part of the ADR equation. Consequently, another method, more precisely the framework, for solution of advection dominated pollutant dispersion model was being developed. Although the ELLAM is much more complicated than the Walcek's method, it has the advantage of incorporating of the other part, diffusion or reaction and the very important boundary conditions, mostly these near the ground. ELLAM has many variants that were mostly adapted to water fluid environments were the behaviour of the flows is different from the atmospheric turbulent flows.

In this paper the form of ELLAM was tested against the Walcek's method. Beside the artificial tests the real wind models with the truly measured data from the performed experiments were chosen for the schemes evaluation.

The paper is organized as follows. In the following section the general and the used ELLAM scheme are outlined. The scheme evaluation including the test inputs and methodology is then described in the ELLAM Evaluation section. Some conclusion and the future work is stated at the end of the paper.

II. ELLAM

ELLAM method is based on a philosophy of algebraic theory by [3]. In this theory, the test functions are used to define the weak form of the governing equation. In the

following subsections the general framework as well as the modified version used in this work is presented.

A. General ELLAM Framework

The procedure of the general ELLAM framework is based on the idea of conversion of equation 1 to the corresponding weak formulation. This is done by multiplying of the governing equation by the test function. Next the whole equation is integrated using Green's formula.

The obtained equation contains several integral terms that have to be evaluated. If we choose the proper form of the test function we can get rid of the one of the integral. This leads from the adjoint equation of equation 1 [4].

The integrals are evaluated analytically for the simple cases, however for practical problems one has to use the numerical approximation. The time discretization is usually done by backward Euler or Runge-Kutta methods. It is an advantage of the scheme that ELLAM can be used generally together within finite difference, finite volume or finite element approaches. The other advantage of the ELLAM framework is its ability to naturally incorporate the boundary conditions. They simply appear as other integral terms in the equation.

The important part of the ELLAM scheme is the accurate characteristic tracking of the points. The problem of characteristics tracking is described by the ordinary differential equations, thus the solution can be obtained by various numerical methods. The more details on general ELLAM method can be found for example in [4].

B. The Presented ELLAM Implementation

Our current implementation of ELLAM framework is in two dimensions (Ω), it is designed for the advection-diffusion equation and for a rectangular grid. We came out from the work of [5] where the space discretization is based on finite element method. The governing equation is:

$$\frac{\partial c}{\partial t} + \nabla\left(\vec{u}c - D\nabla c\right) = f(\vec{x},t), \qquad \vec{x} \in \Re^2, t > 0 \tag{2}$$

where f is the function of the source of the pollution and all other variables have the same meaning as in equation 1. The resulting weak formulation for the specified time t_n after multiplication by test function $w(\vec{x},t_n)$ and applying of Green's formula is:

$$\begin{aligned} &\int_\Omega (cw)(\vec{x},t_n)dx + \int_{J_n}\int_\Omega (D\nabla c)\nabla w dx dt \\ &+ \int_{\Gamma_n} (\vec{u}c - D\nabla c)\cdot\vec{n} w dy dt = \\ &\int_\Omega (cw)(\vec{x},t_{n-1}^+)dx + \int_{\Sigma_n} f(\vec{x},t)dx dt \end{aligned} \tag{3}$$

where $\vec{n}$ is the normal outward unit vector from the element $dydt$, J_n is a time domain, Γ_n is a boundary domain, $\Sigma_n = \Omega \times J_n$, $dydt \in \partial\Omega \times J_n$ and $w(\vec{x},t_{n-1}^+) = \lim_{t\to t_{n-1}^+} w(\vec{x},t)$.

The second integral on the left hand side of the equation 3 is a diffusion term, the third integral is a boundary term and the second integral on the right hand side is a source term.

To evaluate the equation 3, the following procedure is done. The test function was chosen as piecewise linear as is usual for the ELLAM scheme [4]. The terms with integration by time are approximated by backward Euler method. The remaining integrals can be evaluated by numerical integration using for example Gaussian quadrature with appropriate integration points. It remains to evaluate the equation $w(\vec{x},t_{n-1}^+) = \lim_{t\to t_{n-1}^+} w(\vec{x},t)$. This problem leads to the solution of the ordinary differential equation back in time. The common integration methods such as Euler can be used. In our case we decided to use the 4th order Runge-Kutta method. It is a trade-off between speed and accuracy and it behaved very well in cases of the performed experiments.

The last think to explain is the space discretization. We used the rectangular grid of points and the standard FEM process. The equation 3 has to be solved on the whole domain, therefore the elements, on which the approximation of the unknown function c is defined, have to be assembled together. This leads to the system of algebraic equations that has to be solved at each time step.

III. ELLAM EVALUATION

The most of the tests done for the numerical schemes uses the artificial wind velocity fields and then they are evaluated using them. The wind velocities do not have to represent the performance of the scheme in concrete practical applications. Therefore, we have decided to do the tests of the ELLAM scheme against the scheme presented by [2] on the wind velocity fields based on real wind models and real measured data.

A. Used Wind Model

The used wind model is in the stationary form and it depends on the z (height) variable.

The special coefficients are needed in order to calculate the wind speed at given position. These are Monin-Obukhov length (L), roughness length (z_0), von Kármán constant (k) and friction velocity (u_*) [6; 7]. Then, the wind speed is given by:

$$U_z = \frac{u_*}{k}\left[\ln\left(\frac{z}{z_0}\right) - \Psi_m\left(\frac{z}{L}\right) + \Psi_m\left(\frac{z_0}{L}\right)\right], \qquad z \le z_b \tag{4}$$

$$U_z = U_z(z_b), \qquad z > z_b \tag{5}$$

The coefficient $z_b = \min(|L|, 0.1h)$, where the h is a height of the unstable boundary layer.

The function Ψ_m is of the form:

$$\Psi_m = 2\ln\left(\frac{1+A}{2}\right) + \ln\left(\frac{1+A^2}{2}\right) - 2\tan^{-1}(A) + \frac{\pi}{2} \quad (6)$$

$$A = \left[1 - \left(\frac{16z}{L}\right)\right]^{0.25} \quad (7)$$

B. Experiment Data

The above explained wind model has the specific parameters that differ in cases of different experiments. The experiments done in Copenhagen [8] were chosen to completely describe the wind model.

There were 9 experiments performed in Copenhagen, in which all of the required parameters were measured. The all parameters of the experiments that were used for calculations are shown in Table 1.

TABLE I. THE PARAMETERS OF THE PERFORMED EXPERIMENTS IN COPENHAGEN.

Exp. No.	H_s(m)	h(m)	L(m)	u_*(ms^{-1})	w_*(ms^{-1})	k	z_0(m)
1	115	1980	-37	0,36	1,70	0,4	0,6
2	115	1920	-291	0,73	1,80	0,4	0,6
3	115	1120	-71	0,38	1,10	0,4	0,6
4	115	390	-134	0,38	0,74	0,4	0,6
5	115	820	-456	0,45	2,50	0,4	0,6
6	115	1300	-433	1,05	2	0,4	0,6
7	115	1850	-103	0,64	2,10	0,4	0,6
8	115	810	-56	0,69	2,10	0,4	0,6
9	115	2090	-290	0,75	2,00	0,4	0,6

C. The Experiments

Experiments were done for the all cases of the performed Copenhagen experiments. The space was discretized to 100x100 points with the 40m spacing. The wind velocity field was pre-calculated using the equations 4-7 and the data in Table 1. The initial conditions for the test cases were defined by the initial concentration profile at zero time.

Two cases were tested. The first one has the shape of cone and the second one has the initial shape in form of cylinder. The cylinder case simulates the very sharp edges of concentrations which is very critical for the numerical schemes.

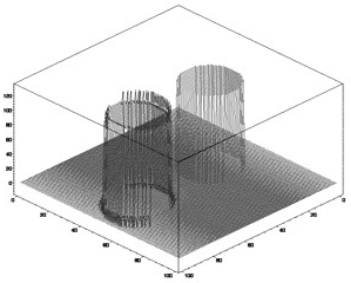


Figure 1. The concentration profiles at the beginning of the simulation (red) and at the end of the simulation (blue) in case of no artificial diffusion addition.

On the other hand the cone shape has the single maximum value of concentration and the numerical schemes have often tendency to smooth it. The diffusion in these test was set to zero, therefore the concentration profile should remain the same at the end of the experiments.

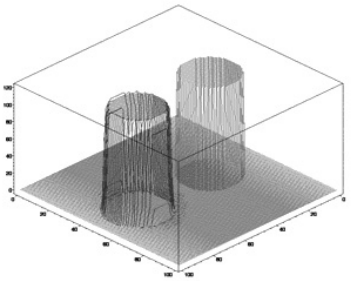


Figure 2. The concentration profiles at the beginning of the simulation (red) and at the end of the simulation (blue) in case of artificial diffusion addition.

D. Oscillations

For the very steep concentration profiles like the cylinder the oscillations in case of ELLAM scheme appear. The situation is shown in Figure 1.

To avoid the oscillations the artificial diffusion was added to the model. As a result, the oscillations disappeared afterwards. The new situation is shown in Figure 2.

E. Error Measurement and Results

Each of the nine experiments was evaluated by several error measurements. The first one is the peak error (Equation 8). It demonstrates how the scheme is able to preserve the level of the concentration with respect to certain profile. The oscillations have the very negative effect to this measure.

$$Err_{peak} = 1 - \frac{Peak_c - Min_c}{Peak_0 - Min_0} \quad (8)$$

$Peak_c$ resp. $Peak_0$ is the peak and Min_c resp. Min_0 is the minimum of calculated resp. precise concentration profile.

The second measure is the distribution error (Equation 9). It describes the difference between the concentration distribution at the beginning and at the end of the simulation. The position of the profile has no influence to the error size.

$$Err_{dist} = 1 - \frac{\sum_{i\in\Omega i}\sum_{j\in\Omega j} Calc_{i,j}}{\sum_{i\in\Omega i}\sum_{j\in\Omega j} exact_{i,j}} \quad (9)$$

Ω_i and Ω_i refers to domain where $Calc_{i,j}$ and $exact_{i,j}$ differs from Min_0.

The last error measurement is referred to the mass conservation principle. In other words the sum of the concentration in the domain at the beginning of the simulation should be the same as at the end of the simulation. For the completeness of the evaluation we measured the calculation times of the schemes as well.

The results of the first set of tests are summarized in Table 2. ELLAM scheme gained the smaller peak error than the Walcek scheme, the similar distribution and final mass error. The calculation times was nearly the same, however one should be noted that the used ELLAM scheme that is described in section 2.2 includes the calculation of the diffusion term which was set to zero for the purposes of the error evaluation.

TABLE II. THE RESULTS OF THE PERFORMED EXPERIMENTS IN CASE OF CONE PROFILE.

Cone	Peak error (%)	Distribution error (%)	Final mass error (%)	calculation time (s)
ELLAM	1,280	0,057	0,053	1,894
Walcek	2,779	0,035	0,027	1,901

The second test includes the cylindrical profile of the concentration. As was mentioned in section 3.3, the undesired oscillation due to the very sharp edges can occur during the simulations. Therefore, we performed the tests for the two versions of the ELLAM scheme.

TABLE III. THE RESULTS OF THE PERFORMED EXPERIMENTS IN CASE OF CONE PROFILE.

Cylinder	Peak error (%)	Distribution error (%)	Final mass error (%)	calculation time (s)
ELLAM	10,203	0,833	0,000	1,848
$ELLAM_D$	2,080	4,332	0,000	1,643
Walcek	0,000	3,012	0,000	1,848

The first was with zero diffusion and the second one was with the artificial diffusion added to the model. The results are shown in Table 3, where the scheme with diffusion is marked as $ELLAM_D$. It can be seen that in case of the zero diffusion ELLAM scheme gained relatively big peak error but very low distribution error and vice versa. It can be seen that the trade-off between the two cases should be appropriate. For further work on this problem we can be inspired by the [9] where the authors developed the technique to prevent the oscillation for 1D case. The calculation times are nearly the same as in previous test case.

IV. CONCLUSION AND FUTURE WORK

We used the specific ELLAM scheme to solve the advection-diffusion equation with the parameters of atmospheric wind velocity fields. We performed the experiments and compared the scheme with the used advection integration scheme designed by Walcek et al. From the performed experiments it is obvious that the ELLAM scheme is more than equal competitor to the state of the art method. It comes from the fact that the errors as well as the calculation time were similar and the ELLAM framework contained the diffusion computation as well.

For the further development we plan firstly to try to avoid oscillation in the scheme together with preserving the low distribution error. The next step would be the testing of the ELLAM scheme with the complete advection-diffusion atmospheric model and its evaluation against the real collecting data measurement. From the effectiveness point of view we are planning to rewrite the ELLAM code to the parallel platform using the very popular OpenCL framework [10].

ACKNOWLEDGMENT

This work is partially supported by the BUT FIT grants ”Information Technology in Biomedical Engineering”, GA102/09/H083 and “Advanced secured, reliable and adaptive IT”, FIT-S-11-1 and the research plan ”Security-Oriented Research in Information Technology”, MSM0021630528.

REFERENCES

[1] D. Lanser, and J.G. Verwer, Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modelling, in J. Comput. Appl. Math, 1999.

[2] C.J. Walcek, and N.M. Aleksic, A simple but accurate mass conservative, peak-preserving, mixing ratio bounded advection algorithm with fortran code, in Atmospheric Environment, 1998.

[3] I. Herrera, Boundary Methods: An Algebraic Theory. Pitman Publishing Limited, New York, 1984.

[4] T.F. Russel, and M.A. Celia, An overview of research on Eulerian–Lagrangian localized adjoint methods (ELLAM), in Advances in Water Resources 25, 2002.

[5] J. Liu, The White Paper on ELLAM Implementation in C++. 2009.

[6] S. Wortmann, M.T. Vilhena, D.M. Moreira, and D. Buske, A new analytical approach to simulate the pollutant dispersion in the PBL, in Atmospheric Environment 39, Elsevier, 2005.

[7] A.G. Ulke, New turbulent parameterization for a dispersion model in the atmospheric boundary layer., in Atmospheric Environment 34, Elsevier, 2000.

[8] S.E. Gryning, and E. Lyck, The Copenhagen Tracer Experiments: Reporting of Measurements., Risø National Laboratory, 1998.

[9] A. Younes, M. Fahs, P. Ackerer, A new approach to avoid excessive numerical diffusion in Eulerian–Lagrangian methods, in Communications in numerical methods in engineering 24, Wiley, 2008.

[10] Khronos, The OpenCL Specification, version 1.0. Khronos OpenCL Working Group, 2009.

Possibilistic models of hybrid systems with nondeterministic continuous evolutions and switchings

Ievgen Ivanov
Taras Shevchenko
National University of Kyiv, Ukraine
Email: ivanov.eugen@gmail.com

Mykola Nikitchenko
Taras Shevchenko
National University of Kyiv, Ukraine
Email: nikitchenko@unicyb.kiev.ua

Louis Feraud
Paul Sabatier University,
Toulouse, France
Email: louis.feraud@irit.fr

***Abstract*—We propose a new class of hybrid (discrete-continuous) dynamical system models with nondeterministic continuous evolutions and switching between discrete modes. Formally, this class is rather similar to the class of stochastic hybrid systems, but it is based on possibility theory. This approach has an advantage over stochastic models when available statistical information is not sufficient for constructing a reliable stochastic model. For example, it may be useful for modeling human-machine-environment systems, because, as it has been argued in the literature, possibility theory describes many aspects of human behavior better than probability theory. In this work we present a motivating example, give a definition and trace semantics of our systems, consider reachability problem for a subclass of them and propose a method to tackle this problem.**

I. Introduction

Hybrid systems [1], [2], [3] are dynamical systems with interacting continuous-time and discrete-event dynamics. Continuous-time dynamics is usually modeled by differential equations and discrete-event dynamics is usually modeled by automata. These systems have a wide range of applications including automation, process control, communication, mechatronics, transportation systems, robotics, real-time software and other fields. In many applications, a hybrid system represents a continuous plant and a discrete controller that switches between modes of the plant. As a mathematical model, a hybrid system is often considered as idealization of a real cyber-physical system. But the adequacy of such modeling may become questionable, because switching conditions and differential equations may describe dynamics imprecisely, values of some parameters may be unknown, etc. For these reasons extensions of a hybrid system model which allow uncertainties may be useful. Researches have proposed several different ways to incorporate uncertainty into hybrid systems [2], [3], [4], [5]: nondeterministic and stochastic hybrid systems, systems with random structure, etc. Most of these approaches can be classified according to chosen places of uncertainty in a hybrid system (continuous dynamics, occurrence of discrete events, jumps in continuous states after discrete events) and a kind of an underling uncertainty theory. Many models are based on probability theory. However, this is not the only available variant.

In this paper we propose a new model of a hybrid system with uncertain switching. This model is based on possibility theory [6], [7], [8], [9] . We argue that this model is well-suited for modeling human-machine systems (e.g. a driver-vehicle system).

The paper is organized in the following way: in section 2 we consider a problem of modeling driver-vehicle system and propose a possibilistic model of hybrid system for this problem (on informal level); in section 3 we recall necessary notions of possibility theory; in section 4 we formally define a subclass of (possibilistic) systems with uncertain switching and investigate basic properties of systems of this class; in the last section we give conclusions and describe future plans.

II. Motivating example

Consider a problem of modeling human behavior in a driver-vehicle-environment system [11], [12], [13]. Interest in this problem comes from applications in safety analysis (driver behavior is known to be the dominant factor in traffic safety), intelligent driver assistance systems, etc. [13]. The history of driver's behavior modeling can be traced back to theoretical studies of driving by J. Gibson, L. Crooks (1938) [14]. The problem proved to be difficult and approaches originating from different fields (psychology, control theory, etc.) and with different aims were proposed [11]. But it is generally accepted, that driving task can be considered at strategic, tactical, operational levels [15] and that comprehensive driving models should take into account these levels. Driver's behavior is usually represented by rules describing actions, which driver takes in response to a driving situation (collection of external factors, e.g. road and weather condition, distances to other vehicles) to achieve the purpose of the trip. Strategic behavior is responsible for a trip route, preferred travel lane, etc. Tactical behavior determines driver's vehicle-maneuvering actions, e.g. lane changing for danger avoidance. Operational behavior consists in a (mostly continuous) lateral control and longitudinal speed control (achieved by changing between gas and break at discrete times).

We consider an example of a driver's behavior model of tactical / operational level, which is focused on inter-vehicle

distance maintenance [16]. Suppose that the driver of a running vehicle behaves rationally and tries to avoid crashes and keep safe distances to other vehicles. Under these assumptions in [16] the driver's procedural knowledge is modeled by a set of IF/THEN rules.

Consider a simplified subset of such rules (where a safe following distance can be determined using a so-called "three seconds rule" [16]):

- IF the distance to the preceding vehicle is less than safe following distance THEN decelerate to maintain safe following distance.
- IF the distance to the preceding vehicle is equal to the safe following distance THEN maintain current distance.
- IF the preceding vehicle's braking light flashes AND the distance to the preceding vehicle is equal to safe following distance THEN brake and decelerate immediately.
- IF the preceding vehicle's braking light flashes AND the distance to the preceding vehicle is less than safe following distance AND the vehicle behind is traveling not too closely THEN brake.

This set of rules is given just for illustration purposes.

Suppose that we want to apply these rules to construct a mathematical model of a driver-vehicle-environment system and that a vehicle-environment model is already available. Suppose that the vehicle-environment model outputs a (dynamically changing) value y which represents elements of the vehicle-environment model observable by the driver (e.g. distance to the preceding vehicle). For the purpose of a high-level modeling, it may be possible to ignore details of driver's perception (like in many control-theoretic and stochastic driver behavior models). In this case, we can model driver's behavior as a decision procedure based on driving rules and accept, that driver's decisions directly depend on y. However, vagueness of driving rules (e.g. meaning of the conditions like "too close") implies that decisions depend on y non-deterministically and we have to apply some uncertainty theory to describe them. In the literature [10] it is argued, that *possibility theory* is well suited for representing subjective estimations of satisfaction and acceptability (vague threshold values), perception and quantities based on memory (e.g. travel time, distance, appearance), descriptive condition (e.g. traffic congestion, comfort, safety), imprecise values which are hard to measure and summarize (e.g. sight distance, reaction time).

In our case, we deal with notions of subjective perception and acceptability, so we apply possibility theory for modeling uncertainty in driver's decisions. The proposed model of driver's behavior (Fig. 1) has a form of an oriented graph of "driving modes".

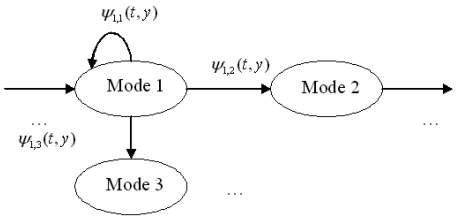


Fig. 1. A model of driver's behavior.

Arcs ("transitions between modes") are labeled with real-valued functions $\psi_{i,j}$ which represent uncertain switching conditions. These conditions may depend on time t and observable vehicle's state y. For each t and y the meaning of $\psi_{i,j}(t,y) \in [0,1]$ is a level of (conditional) possibility of transition from the mode i to the mode j if the current mode is i, time is t and an observable state is y. Note that if $\psi_{i,j}$ are constant functions, we get a model similar to possibilistic Markov chains [17], [18], [19].

For example, we can interpret the "Mode 1" in Fig. 1 as "Normal (constant-speed) driving', "Mode 2" as "Decelerate" and y as a distance to the preceding car. Then we can represent uncertain switching condition "Preceding car is too close" as a specific function $\psi_{1,2}$.

To determine semantics of our model of driver's behavior, we have to link it with vehicle-environment model. Usually a vehicle-environment dynamics corresponding to a given driving mode can be modeled by a system of differential equations (possibly with disturbances / uncertainties) [11]. However, in this paper we do not consider disturbances and uncertainties in differential equations. The resulting driver-vehicle-environment model has a form shown in Fig. 2, where $y_{all} \in \mathbb{R}^d$ denotes vehicle-environment (internal) state and $y \in \mathbb{R}^{d'}$ is an observable state (the relation between y and y_{all} is given by projection-like functions g_i). For technical reasons it is convenient to allow instant changes ("jumps") in y_{all} at mode switching points (which are denoted by expressions like $y_{all} := h_{1,2}(t, y_{all})$).

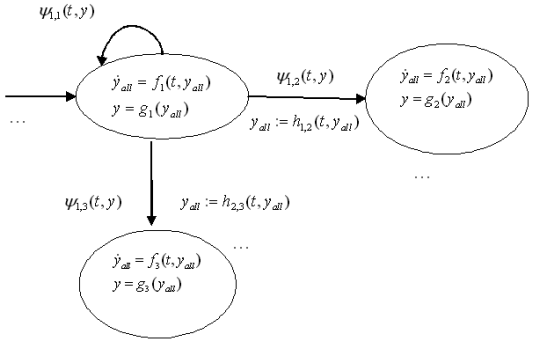


Fig. 2. A driver-vehicle-environment model.

We call the obtained driver-vehicle-environment model a *hybrid system with uncertain switching*. In some aspects it is similar to the notion of hybrid automaton [1], [2], however, mode switching is uncertain and is modeled in possibilistic framework.

Now we informally explain semantics of such a system. Denote as G a labeled graph on Fig 2. Let I be a finite set of modes (nodes of G). First, let us ignore arc labels $\psi_{i,j}$ in this graph and define a *trace* of the system as a triple $(\bar{q}, \bar{y}_{all}, \bar{y})$ of a piecewise-constant function $\bar{q} : \mathbb{R} \to I$ (a "mode trace"), a piecewise-continuous function $\bar{y}_{all} : \mathbb{R} \to \mathbb{R}^d$ (a "state trace") and a function $\bar{y} : \mathbb{R} \to \mathbb{R}^{d'}$ (an "observable state trace"), such that $(\bar{q}(0), \bar{y}_{all}(0)) \in A_0$ (where 0 is an initial time moment and A_0 is a chosen set of initial modes/states) and there exists a finite or infinite sequence pairs of time moments and modes $(t_0, q_0), (t_1, q_1), (t_2, q_2), ...$ of the length $N \geq 1$ ($N = +\infty$ if the sequence is infinite) such that $0 = t_0 < t_1 < t_2 < ...$, the sequence $t_0, t_1, t_2, ...$ is unbounded if $N = +\infty$, $(\bar{q}(0), q_0)$ is an arc in G, for each $k = 1, 2..., N-1$, (q_{k-1}, q_k) is an arc in G, and for each $k < N$ functions $\bar{q}$, $\bar{y}_{all}$, $\bar{y}$ satisfy equations $\bar{q}(t) = q_k$, $\bar{y}_{all}(t) = z_k(t)$, $\bar{y}(t) = g_{q_k}(\bar{y}_{all}(t))$ on the set $(t_k, t_{k+1}]$ if $k+1 < N$, or $(t_k, +\infty)$ if $k+1 = N$, where z_k is a solution of the initial value problem $z'_k(t) = f_{q_k}(t, z_k(t)), t \in [t_k, t_{k+1}]$ (or $t \in [t_k, +\infty)$ if $k+1 = N$), $z_k(t_k) = h_{q_{k-1}, q_k}(\bar{y}_{all}(t_k))$ if $k > 0$, or $z_k(t_k) = h_{\bar{q}(0), q_k}(\bar{y}_{all}(t_k))$ if $k = 0$. Note that for simplicity and unlike hybrid automata semantics, this definition excludes multiple mode switchings at one time instance, Zeno-like behaviors, etc. [1].

Let Tr be the set of all traces. Semantics of a hybrid system with uncertain switching is a *possibility distribution* on traces, i.e. a (total) function $\pi : Tr \to [0, 1]$ which estimates possibility level of each trace. If we knew this distribution, we would be able to compute the following quantities: $\varphi_{ij}(t, y) = \sup\{\pi((\bar{q}, \bar{y}_{all}, \bar{y}))|\ (\bar{q}, \bar{y}_{all}, \bar{y}) \in Tr$ and $\bar{q}(t) = i$, $q(t+) = j$, $\bar{y}(t) = y\ \}$ for each $i, j \in I$, $t \geq 0$ and $y \in \mathbb{R}^{d'}$. In possibility theory the value $\varphi_{ij}(t, y)$ can be interpreted as a possibility level of the event "the system switches from i to j at moment t in the observable state y". Now recall that the value $\psi_{ij}(t, y)$ informally means a conditional possibility of the same event under the condition that the system is in the state i at moment t. Intuitively, the value $\psi_{ij}(t, y)$ can be the same or larger than $\varphi_{ij}(t, y)$, because to make a transition from i to j at time t, the system should reach i at time t, but this can be implausible or even impossible. Hence we postulate the inequality $\varphi_{ij}(t, y) \leq \psi_{ij}(t, y)$ for all i, j, t, y. Now we propose to consider this inequality as a definitional property of π. In general case it has no unique solution (if we consider π an unknown), but we can choose the (point-wise) maximal solution as the semantics of the system (principle of minimum specificity [7]). Informally, this solution gives the best estimate of the possibilities of traces under the chosen assumptions.

It is easy to see that quantities $\varphi_{ij}(t, y)$ express important safety properties of the system, e.g. we can use them to obtain upper bounds for possibility levels of some unwanted actions (mode switchings) and states. Therefore to be able to solve safety analysis problems we should have a method of computation of the values $\varphi_{ij}(t, y)$ from the given functions ψ_{ij}. To make explicit the dependence between these values and functions, note that if we (point-wise) increase π, then the values $\varphi_{ij}(t, y)$ also increase for each fixed i, j, t, y. Hence $\varphi_{ij}(t, y)$ are solutions of the following optimization problem (where π is variable distribution):

1) $\varphi_{ij}(t, y) \to \max$ (for each i, j, t, y)
2) $\varphi_{ij}(t, y) \leq \psi_{ij}(t, y)$ for all i, j, t, y
3) $\varphi_{ij}(t, y) = \sup\{\pi((\bar{q}, \bar{y}_{all}, \bar{y}))\ |\ (\bar{q}, \bar{y}_{all}, \bar{y}) \in Tr$ and $\bar{q}(t) = i$, $q(t+) = j$, $\bar{y}(t) = y\ \}$

We have outlined the general idea of a hybrid system with uncertain switching. To make the first step to formalization of these systems, let us consider their subclass ("simple systems with uncertain switching"), determined by the following constraints:

- Internal state y_{all} is observable, i.e. $y = y_{all}$;
- There are no jumps in state y, i.e. $h_{i,j}(t, y) = y$;
- Functions $\psi_{i,j}$ (uncertain switching conditions) depend only on time t.

In the rest of the paper we will give a rigorous definition and semantics of simple systems with uncertain switching, and study their properties. We plan to formalize and investigate a general class of hybrid systems with uncertain switching in the forthcoming articles.

III. Possibility theory and Markov-like processes

We use the following framework of (quantitative) possibility theory [6], [7], [8], [9]. Let X be a space of atomic events.

Let $\Pi : 2^X \to [0, 1]$ be a (total) possibility measure, i.e.

$$\Pi(\bigcup\nolimits_k A_k) = \sup_k \Pi(A_k)$$

for any family $\{A_k\}_k$ of subsets of X (events), and let $N : 2^X \to [0, 1]$ be a necessity measure, i.e.

$$N(\bigcap\nolimits_k A_k) = \inf_k N(A_k)$$

for any family $\{A_k\}_k$. We assume that $N(A) = 1 - \Pi(\neg A)$ for all $A \subseteq X$ (here $\neg A$ denotes complement of a set), $\Pi(X) = 1$ and $N(X) = 1$.

Let T be the timeline $[0, +\infty)$ and Y be a set. We use the following conventions: if $pred$ is some predicate on X, then $\Pi\{x : pred(x)\}$ denotes $\Pi(\{x \in X : pred(x)\})$; a variable t denotes time and ranges over T.

Under our assumption of totality of measures we will use the following terminology:

- a *possibilistic variable* is a total function $X \to Y$;
- the *distribution* of a possibilistic variable $\xi : X \to Y$ is a mapping $y \mapsto \Pi\{x : \xi(x) = y\}$;
- a *process* is an arbitrary partial function $T \times X \to Y$;
- a *trajectory* of a process $p : T \times X \to Y$ is a function $t \mapsto p(t, x)$ for an arbitrary fixed x;
- the *distribution of a process* $p : T \times X \to Y$ is a function $F_p : 2^{T \to Y} \to L$ such that

$$F_p(q) = \Pi\{x : \forall t\, p(t, x) = q(t)\}$$

where $q : T \to Y$, i.e. F_p gives a possibility of q to be a trajectory of p;

- an α-*trajectory* of p (where $\alpha \in [0, 1)$) is a mapping $q : T \to Y$ such that $F_p(q) > \alpha$, i.e. q is a trajectory of p with possibility greater than α.

To rigorously define our possibilistic model we need a possibilistic counterpart of a (stochastic) Markov process. In the context of possibility theory Markov processes and related notions were investigated in several works e.g. [17], [18], [19], [20], [21]. However, these notions are not well suited for our purposes, so we introduce a new notion of possibilistic Markov process which we call Markov-like process to avoid confusion.

Let q_1,q_2 be trajectories of a process p such that $q_1(t^*) = q_2(t^*)$. Then the *crossover trajectories* of q_1 and q_2 at t^* are functions $\bar{q}_1$, $\bar{q}_2 : T \to Y$ such that

- $\bar{q}_1(t) = q_1(t)$ if $t \le t^*$ and $\bar{q}_1(t) = q_2(t)$ if $t \ge t^*$,
- $\bar{q}_2(t) = q_2(t)$ if $t \le t^*$ and $\bar{q}_2(t) = q_1(t)$ if $t \ge t^*$.

Informally, $\bar{q}_1$ and $\bar{q}_2$ are obtained by gluing together parts of q_1 and q_2 before and after t^*.

We say that a process p has *Markov-like* property if for any α-trajectories q_1,q_2 of p such that $q_1(t^*) = q_2(t^*)$ for some t^*, the crossover trajectories of q_1 and q_2 at t^* are α-trajectories of p. Consider Markov-like processes with piecewise-constant trajectories. We call a Markov-like process p a *Markov-like jump process* if for each trajectory q of p:

1) q is piecewise constant and right-continuous;
2) $\Pi\{x : \forall t\, p(t, x) = q(t)\} =$
$= \lim\limits_{t^* \to +\infty} \Pi\{x : \forall t \le t^*\, p(t, x) = q(t)\}$

The second condition ensures that possibility of a trajectory is determined by possibilities of its bounded parts.

An *unconditional transition distribution* of a Markov-like jump process $p : T \times X \to I$ (where I is a non-empty state space) is an indexed family $(\varphi_{i,j})_{i,j \in I}$ of functions defined as

$$\varphi_{i,j}(t) = \Pi\{x : p(t, x) = i, \lim_{\tau \to t+} p(\tau, x) = j\}, \quad t \in T$$

i.e. $\varphi_{i,j}(t)$ is a possibility of transition from i to j at the time moment t.

The following simple lemma shows that the distribution of a Markov-like jump process is uniquely determined by its unconditional transition distribution.

Lemma 1: If p is a Markov-like jump process, then

$$F_p(q) = \Pi\{x : \forall t\, p(t, x) = q(t)\} = \inf_{t \in T} \varphi_{q(t), q(t+)}(t),$$

for each piecewise constant right-continuous function $q : T \to Y$ (where $q(t+)$ denotes right limit).

Note that not every family of functions is an unconditional transition distribution.

Lemma 2: A family $(\varphi_{i,j})_{i,j \in I}$ is an unconditional transition distribution of some Markov-like jump process if and only if the following conditions are satisfied:

1) $\sup\limits_{i,j \in I} \varphi_{i,j}(t) = 1$ for all $t \in T$;
2) $\varphi_{i,j}(t_0) = \sup\{ \inf\limits_{t \in T} \varphi_{q(t), q(t+)}(t) \mid$ a function $q : T \to I$ is piecewise constant, right-continuous and $q(t_0) = i$, $q(t_0+) = j\}$, for each $i, j \in I$ and $t_0 \in T$.

Usually (as in our motivating example) we do not have an unconditional transition distribution as a process specification. Instead, we have a family of conditional possibilities

$$\psi_{i,j}(t) = \Pi\{x : p(t, x) = i, \lim_{\tau \to t+} p(\tau, x) = j | p(t, x) = i\},$$

i.e. a possibility of a transition from i to j at a time moment t, if the process is in the mode i. Here we have the following problem: there is no universally accepted formal definition of the conditional possibility. Several authors proposed different approaches to such a definition [8]. We propose to overcome this problem using the following observation: we are dealing with a conditional possibility of the form $\Pi(A|B)$, where $A \subseteq B$, and in this case most definitions of conditional possibility imply the property $\Pi(A|B) \ge \Pi(A)$. We take this relation as an axiomatic definition of conditional possibility if $A \subseteq B$. Note that $\Pi(A|B)$ and $\Pi(A)$ do not determine each other uniquely.

We propose the following specification mechanism for Markov-like jump processes. Suppose that we know (informal) conditional possibilities of transitions $(\psi_{i,j})_{i,j \in I}$ and we want to construct a Markov-like jump process distribution from this knowledge. To do this, we try to find an unconditional transition distribution $(\varphi_{i,j})_{i,j \in I}$ from the following conditions:

1) $\varphi_{i,j}(t) \le \psi_{i,j}(t)$, i.e. conditional possibilities are upper bounds for unconditional possibilities;
2) $(\varphi_{i,j})_{i,j \in I}$ is the greatest (the least specific) among all unconditional transition distributions $(\varphi'_{i,j})_{i,j \in I}$ satisfying $\varphi'_{i,j}(t) \le \psi_{i,j}(t)$.

Then $(\varphi_{i,j})_{i,j \in I}$ determines distribution of a Markov-like jump process.

More formally, we call a family of functions $(\psi_{i,j})_{i,j \in I}$ (where $\psi_{i,j} : T \to [0, 1]$) an *upper transition distribution*, if there exists an unconditional transition distribution $(\varphi_{i,j})_{i,j \in I}$ such that $\varphi_{i,j}(t) \le \psi_{i,j}(t)$ for all $i, j \in I$, $t \in T$, i.e. it is an upper bound for some unconditional transition distribution. In particular, conditional transition possibilities form an upper transition distribution.

Lemma 3: A family $(\psi_{i,j})_{i,j \in I}$ is an upper transition distribution if and only if

$$\sup\{ \inf_{t \in T} \psi_{q(t), q(t+)}(t) \mid \text{ a function } q : T \to I \text{ is piecewise constant and right-continuous} \} = 1.$$

Corollary. If $\max\limits_{j \in I} \psi_{i,j}(t) = 1$ for all $i \in I$ and $t \in T$, then $(\psi_{i,j})_{i,j \in I}$ is an upper transition distribution.

We say that an unconditional transition distribution $(\varphi_{i,j})_{i,j \in I}$ is *generated* by upper transition distribution $(\psi_{i,j})_{i,j \in I}$ if the following conditions are satisfied:

- $\varphi_{i,j}(t) \le \psi_{i,j}(t)$, $i, j \in I$, $t \in T$;
- $\varphi'_{i,j}(t) \le \varphi_{i,j}(t)$, $i, j \in I$, $t \in T$, for each unconditional transition distribution $(\varphi'_{i,j})_{i,j \in I}$ such that $\varphi'_{i,j}(t) \le \psi_{i,j}(t)$, $i, j \in I$, $t \in T$.

Lemma 4: Each upper transition distribution generates a unique unconditional transition distribution.

This lemma implies that we can specify any Markov-like jump process (up to distribution) in the following way: specify an upper transition distribution and find a generated unconditional transition distribution from it.

The main property of this specification mechanism is that if we fix some definition of conditional possibility (such that $\Pi(A|B) \geq \Pi(A)$ whenever $A \subseteq B$) and define an upper transition distribution as a family of conditional possibilities of transitions of some Markov-like jump process p, then the generated unconditional transition distribution gives an upper estimate for the unconditional transition distribution of the process p.

Consider the problem of computation of a generated distribution from a given upper transition distribution. We propose solution to this problem in the case when upper transition distribution belongs to a special class defined below. We argue that this class is sufficient for many practical purposes.

We call a function $f : T \to L$ *piecewise-monotone* if for every $t_0 \in T$ there exists a relatively open (in T) neighborhood O of t_0 such that f is monotone on sets $O \cap [0, t_0)$ and $O \cap (t_0, +\infty)$ (if they are non-empty). An upper transition distribution $(\psi_{i,j})_{i,j \in I}$ is piecewise-monotone if the set I is finite and each function $\psi_{i,j}$ is piecewise-monotone.

The following theorem describes a monotonic iterative method for computing $(\varphi_{i,j})_{i,j \in I}$.

Theorem 1: Let $(\psi_{i,j})_{i,j \in I}$ be a piecewise-monotone upper transition distribution and $(\varphi_{i,j})_{i,j \in I}$ be a corresponding generated unconditional transition distribution. Let $(\psi^n_{i,j})_{i,j \in I}$, $n = 0, 1, 2, ...$ be a sequence of families of functions defined by the following equations:

1) $\psi^{n+1}_{i,i}(t) =$

$$= \inf_{t' < t} \left(\psi^n_{i,i}(t') \vee \sup_{j \in I \setminus \{i\}, \tau \in [t', t)} (\psi^n_{j,i}(\tau) \wedge \psi^n_{j,j}(t')) \right) \wedge \psi^n_{i,i}(t) \wedge \inf_{t' > t} \left(\psi^n_{i,i}(t') \vee \sup_{j \in I \setminus \{i\}, \tau \in (t, t']} (\psi^n_{i,j}(\tau) \wedge \psi^n_{j,j}(t')) \right)$$

   if $n \geq 0$, where where $\vee$ and $\wedge$ denote binary maximum and minimum operations on the segment [0,1];

2) $\psi^{n+1}_{i,j}(t) = \psi^n_{i,i}(t-) \wedge \psi^n_{i,j}(t) \wedge \psi^n_{j,j}(t+)$, $n \geq 0$, $i \neq j$ (here we assume that $\psi_{i,i}(0-) = 1$, where 0 is the initial moment of time).

Then for each i, j, the sequence of function $\psi^n_{i,j}, n \geq 0$ pointwise converges to $\varphi_{i,j}$.

Note that if the condition $\max_{j \in I} \psi_{i,j}(t) = 1$ is satisfied, then the recurrent equations for $(\psi^n_{i,j})_{i,j \in I}$ can be simplified.

IV. Systems with uncertain switching and a reachability problem

Let I be a non-empty finite set of states, $T = [0, +\infty)$, and $p : T \times X \to I$ be a Markov-like jump process. Let $f_i : T \times \mathbb{R}^d \to \mathbb{R}^d$, $i \in I$ be a family of functions.

We call *a (simple) system with uncertain switching* (SUS) an equation of the form

$$\dot{y}(t, x) = f_{p(t,x)}(t, y(t, x)) \tag{1}$$

Let us denote $X_+ = \{x \in X | \Pi\{x\} > 0\}$, i.e. the set of atomic events of positive possibility.

We call a process $y : T \times X \to \mathbb{R}^d$ a *solution* of SUS (1) if for any fixed $x \in X_+$ the trajectory $t \mapsto y(t, x)$ satisfies equation (1) in sense of Caratheodory, i.e. is absolutely continuous on every compact segment in T and satisfies (1) almost everywhere (with respect to Lebesgue measure). Note that for simplicity we do not use the notions like hybrid execution and hybrid time in this definition [1], [2].

Let $\alpha \in [0, 1)$. We call a function $\bar{y} : T \to \mathbb{R}^d$ a (complete) *α-trajectory* of SUS (1) if $\bar{y}$ is an α-trajectory of some solution of (1). Consider an initial condition

$$y(0, x) = y_0, \quad x \in X_+ \tag{2}$$

We say that the problem (1)-(2) has a *unique solution* (up to trajectories of the possibility zero) if every two solutions of (1) which satisfy (2) coincide on the set $T \times X_+$.

The following theorem is an adaptation of Caratheodory existence theorem [22].

Theorem 2: Suppose that the following conditions are satisfied:

1) for each $i \in I$ and $t \in T$, the function $y \mapsto f_i(t, y)$ is defined and continuous on $\mathbb{R}^d$, and for each $y \in \mathbb{R}^d$, the function $t \mapsto f_i(t, y)$ is measurable;
2) for each $i \in I$ there exists a function $h_i : T \to \mathbb{R}_+$, which is bounded on every bounded segment in $\mathbb{R}$, such that
$$\|f_i(t, y)\| \leq h_i(t)(1 + \|y\|)$$
for all $t \in T$, $y \in \mathbb{R}^d$, where $\mathbb{R}_+ = [0, +\infty)$ and $\|\cdot\|$ denotes Euclidean norm);
3) for each $i \in I$ there exists a function $L_i : T \to \mathbb{R}_+$ (Lipschitz constant), which is bounded on every bounded segment in $\mathbb{R}$ and
$$\|f_i(t, y_1) - f_i(t, y_2)\| \leq L_i(t)\|y_1 - y_2\|$$
for all $y_1, y_2 \in \mathbb{R}^d$ (Lipschitz continuity).

Then for each $y_0 \in \mathbb{R}^d$ the problem (1)-(2) has a unique solution.

Let $(\varphi_{i,j})_{i,j \in I}$ be an unconditional transition distribution of a Markov-like jump process p.

Theorem 3: Suppose that conditions of the theorem 2 are satisfied. Then a function $\bar{y} : T \to \mathbb{R}^d$ is an α-trajectory of (1) if and only if there exists a piecewise constant and right-continuous function $q : T \to I$ such that $\inf_{t \in T} \varphi_{q(t), q(t+)}(t) > \alpha$ and $\bar{y}$ satisfies equation $\dot{y}(t) = f_{q(t)}(t, y(t))$ on T in sense of Caratheodory.

One of the basic analysis problems for SUS is a *reachability problem*: find those states, which are reachable with (at least) given level of possibility. Formally, for any set $Y_0 \subseteq \mathbb{R}^d$ and $\bar{t} \in T$ let us define a *closure of an α-reachable set*:

$cReach^\alpha(Y_0, \bar{t}) = cl(\{\bar{y}(\bar{t}) | \ \bar{y} : T \to \mathbb{R}^d$ is an α-trajectory of SUS (1) and $y(0) \in Y_0\})$, where $cl(\cdot)$ denotes closure of a subset of $\mathbb{R}^d$, i.e. $cReach^\alpha_*(Y_0, \bar{t})$ is a closure of the set of points which can be reached by α-trajectories of SUS (1) at the moment of time $\bar{t}$ from the set Y_0.

We will use the following lemma to describe $cReach^\alpha_*(Y_0, \bar{t})$:

Lemma 5: Let I be a finite set, $(\varphi_{i,j})_{i,j\in I}$ be a piecewise-monotone transition distribution and $\alpha \in [0,1)$. Then there exists an increasing sequence of moments of time $\tau_0, \tau_1, \tau_2, ... \in T$ such that for each $k = 0, 1, 2, ...$ and $i, j \in I$ the function $\varphi_{i,j}$ is monotonous on (τ_k, τ_{k+1}) and either $\varphi_{i,j}(t) > \alpha$ for all $t \in (\tau_k, \tau_{k+1})$ or $\varphi_{i,j}(t) \leq \alpha$ for all $t \in (\tau_k, \tau_{k+1})$.

Suppose that $\alpha \in [0,1)$ is fixed and a sequence $\tau_0, \tau_1, \tau_2, ...$ described in the lemma 5 is given. Let us denote by I^+ the set of non-empty words (finite strings) in alphabet I. For each $i, j \in I$ and $t_0, t_1 \in T$, such that $t_0 < t_1$, denote:

$LS^{\alpha}_{i,j}(t_0, t_1) = \{ii_1i_2...i_n \in I^+ \mid n \geq 1, \quad \varphi_{i,i_1}(t_0) > \alpha, \quad \varphi_{j,j}(t_1-) > \alpha, \quad i_n = j, \quad (i_l, i_{l+1}) \in H^{\alpha}(t_0, t_1)$ for each $l = \overline{1, n-1}\}$, where

$$H^{\alpha}(t_0, t_1) = \{(i, j) \in I \times I \mid \forall t \in (t_0, t_1)\ \varphi_{i,j}(t) > \alpha\}.$$

Then $LS^{\alpha}_{i,j}(t_0, t_1)$ is a regular language in the alphabet I.

For any formal language $L \subseteq I^+$, moments $t_0 < t_1$, and a set $Y_0 \subseteq \mathbb{R}^d$ let us define:

$reach(L, t_0, Y_0, t_1) = \{\bar{y}(t_1) \mid \bar{y} : [t_0, t_1] \to \mathbb{R}^d$ is a function such that $\bar{y}(t_0) \in Y_0$ and there exists a piecewise-constant function $q : [t_0, t_1] \to I$ and time moments $\bar{t}_0, ..., \bar{t}_n \in T$ such that $t_0 = \tau_0 < \tau_1 < ... < \tau_{n-1} < \tau_n = t_1$, $q(t) = i_{k+1}$ for all $t \in (\tau_k, \tau_{k+1})$,$k = \overline{0, n-1}$, and $\bar{y}$ satisfies equation $\dot{y}(t) = f_{q(t)}(t, y(t))$ in the sense of Caratheodory$\}$, i.e. the set of points which are reachable from Y_0 by means of switching sequences described by words in L.

Also let us define an indexed family of sets:

- $Y^0_{i,j} = Y_0$ if $i = j$,
- $Y^0_{i,j} = \emptyset$ if $i \neq j$;
- for each $i, j \in I$ and $k \geq 1$:

$$Y^k_{i,j} = \bigcup\nolimits_{l\in I} reach(LS^{\alpha}_{l,j}(\tau_{k-1}, \tau_k), \tau_{k-1}, Y^{k-1}_{i,l}, \tau_k)$$

The following theorem characterizes α-reachable points of the phase space for SUS.

Theorem 4: Let $\bar{t} \in [\tau_n, \tau_{n+1})$ for some $n \geq 0$, and

$$J_0(\bar{t}) = \{j \in I \mid \max_{j'\in I} \varphi_{j,j'}(\bar{t}) > \alpha\}.$$

Then the following properties are satisfied:

1) If $\bar{t} = \tau_n$, then

$$cReach^{\alpha}(Y_0, \bar{t}) = \bigcup\nolimits_{i\in I, j\in J_0(\bar{t})} cl(Y^n_{i,j});$$

2) If $\bar{t} \in (\tau_n, \tau_{n+1})$, then $cReach^{\alpha}(Y_0, \bar{t}) =$

$$= \bigcup\nolimits_{i,l\in I, j\in J_0(\bar{t})} cl(reach(LS^{\alpha}_{l,j}(\tau_n, \bar{t}), \tau_n, Y^n_{i,l}, \bar{t})).$$

V. Conclusion

We have considered a possibilistic approach to modeling uncertainty in discrete-continuous hybrid systems and outlined potential applications of this approach in driver behavior modeling. With this approach we have proposed a new class of hybrid dynamical systems with uncertain switching between discrete modes. On the basis of possibility theory we have defined a subclass of such systems and investigated its basic properties. A rigorous treatment of a general class of systems with possibilistic switching is a topic of our future research.

Acknowledgment

The authors would like to thank Dr. Didier Dubois and Dr. Martin Strecker of Institut de Recherche en Informatique de Toulouse (IRIT), France for their comments concerning this work during an internal meeting at IRIT on March 28, 2011.

References

[1] R. Goebel, R. Sanfelice, and A. Teel, "Hybrid dynamical systems," *IEEE Control Systems Magazine*, vol. 29 (2), pp. 28–93, 2009.

[2] T. A. Henzinger, "The theory of hybrid automata," in *LICS*, 1996, pp. 278–292.

[3] J. Lygeros and S. Sastry, "The art of hybrid systems," 2001. [Online]. Available: http://robotics.eecs.berkeley.edu/ sastry/ee291e/book.pdf

[4] X. Du, H. Ying, and F. Lin, "Fuzzy hybrid systems modeling," in *Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American*, 2010, pp. 1–6.

[5] H. A. Blom and J. Lygeros, Eds., *Stochastic Hybrid Systems: Theory and Safety Critical Applications*, ser. Lecture Notes in Control and Information Sciences. Springer, 2006, vol. 337.

[6] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 100, pp. 9–34, 1999.

[7] D. Dubois and H. Prade, *Possibility Theory*. New York (NY): Plenum Press, 1988.

[8] G. de Cooman, "Possibility theory I: the measure- and integral-theoretic groundwork," *International Journal of General Systems*, vol. 25, pp. 291–323, 1997.

[9] Y. Belov, O. Bychkov, M. Merkuryev, and A. Chulichkov, "About an approach to modeling fuzzy dynamics," in *Reports of the National Academy of Sciences of Ukraine*, 2006, pp. 14–19.

[10] P. C. S. Kikuchi, "Place of possibility theory in transportation analysis," *Transportation Research*, vol. 40, pp. 595–615, 2006.

[11] P. C. Cacciabue, Ed., *Modelling driver behaviour in automotive environments*. Springer, 2007.

[12] D. Salvucci, "Modeling driver behavior in a cognitive architecture," *Human Factors*, vol. 48, pp. 362–380, 2005.

[13] T. Vaa, "Cognition and emotion in driver behaviour models - some critical viewpoints," in *Proceedings of the 14th ICTCT Workshop*. Caserta, 2001.

[14] K. Gibson and L. Crooks, "A theoretical field-analysis of automobile driving," *The Americal Journal of Psychology*, vol. 51, pp. 453–471, 1938.

[15] J. Michon, "Explanatory pitfalls and rule-based driver models," *Accident Analysis and Prevention*, vol. 21, pp. 341–353, 1989.

[16] Y. Liu and Z. Wu, "Urgent driver behavior modeling in cognitive architecture," in *Proceedings of the 18th ICTCT Workshop*. Helsinki, 2005.

[17] H. Janssen, G. Cooman, and E. E. Kerre, "First results for a mathematical theory of possibilistic markov processes," Aug. 26 1996. [Online]. Available: http://citeseer.ist.psu.edu/50212.html; http://ensmain.rug.ac.be/ gert/publications/markov.ps

[18] H. J. Janssen, G. D. Cooman, and E. E. Kerre, "Coherent models for discrete possibilistic systems," in *ISIPTA*, G. D. Cooman, F. G. Cozman, S. Moral, and P. Walley, Eds., 1999, pp. 189–195. [Online]. Available: http://decsai.ugr.es/ smc/isipta99/proc/074.html

[19] D. Dubois, F. Dupin de Saint Cyr, and H. Prade, "Updating, transition constraints and possibilistic Markov chains," *Lecture Notes in Computer Science*, vol. 945, pp. 263–272, 1995.

[20] C. Joslyn, "On possibilistic automata," *Lecture Notes in Computer Science*, vol. 763, p. 231, 1994.

[21] J. Buckley and E. Eslami, "Fuzzy markov chains: uncertain probabilities," *Math Ware and Soft Computing*, vol. 9, pp. 33–41, 2002.

[22] A. Filipov, "Differential equations with discontinuous right-hand sides," *AMS Trans.*, vol. 42, no. ser. 2, pp. 199–231, 1964.

Simulation Analysis Of Global Optimization Algorithms As Tools For Solving Chance Constrained Programming Problems

Andrzej Z. Grzybowski
Faculty of Mechanical Engineering and Computer Science
Częstochowa University of Technology
Częstochowa, Poland
azgrzybowski@gmail.com

Abstract—**In the paper a chance constrained linear programming problem is considered in the case of join chance constraints with random both left and right hand sides. It is assumed that due to its complex stochastic nature the problem cannot be reduced to any equivalent deterministic problem. In such a case a Monte Carlo method combined with Global Optimization (GO) algorithms are proposed to solve the problem. A performance of various types of GO algorithms as tools for solving such problems are compared via computer simulations. The simulation results are presented and discussed in the paper.**

Keywords- chance constrained programming, Monte Carlo simulations, stochastic search, evolutionary algorithms, anealing type algorithms.

I. INTRODUCTION

Chance Constrained Programming (CCP) or, more general, stochastic programming deals with a class of optimization models and algorithms in which some of the data may be subject to significant uncertainty. Such models are appropriate when data cannot be observed without error or evolve over time and decisions have to be made prior to observing the entire data stream. The concept of CCP was introduced in the classical work of Charnes and Cooper [1]. Now CCP belongs to the major approaches for dealing with random parameters in optimization problems. Typical areas of application are engineering design applications, finance (e.g.[12]), budgeting ([2]) or portfolio analysis [5]. In models built for such real-world problems uncertainties like product demand, cost of supply, price of a final product, demographic conditions, currency exchange rates, rates of return etc. enter the inequalities describing the natural constraints that should be satisfied for proper working of a system under consideration.

Stochastic optimization problems belong to the most difficult problems of mathematical programming. Most of the existing computational methods are applicable only to convex problems. There are, however, many important applied optimization problems which are, at the same time, stochastic and non-convex. Many of them are also multi-extremal. Discussion of various computational aspect of CCP problems can be found in papers [5], [8],[10] or textbooks [4,7].

Our paper is devoted to the linear programming problems which contain random parameters. It is assumed that due to their complex stochastic nature the problems cannot be reduced to any equivalent deterministic problems. To choose the "best solution" we use global optimization algorithms. However it results from the statement of the problem, that the criterion function cannot be expressed by any closed-form mathematical formula. As a consequence we have to use gradient-free optimization methods based on the idea of the stochastic search. In these methods Monte Carlo simulations are the primary source of input information during the search process. Such methods require only few assumptions about the underlying objective functions - they have so-called “black box” character, see [11,13].

In our paper we compare the performance of the most popular stochastic global optimization methods: genetic algorithms, evolutionary search with soft selection and simulated annealing. The performance of the methods as a tools for solving the stochastic linear programming tasks is studied via computer simulations under two different utility criteria.

II. CHANCE CONSTRAINED LINEAR PROGRAMMING PROBLEM

Let us consider a classical (deterministic) linear programming problem:

$$\text{maximize} \quad f(x_1, \ldots, x_n) = c_1x_1+c_2x_2+ \ldots+c_nx_n$$

subject to the constrains (s.t.):

$$a_{i1}x_1+a_{i2}x_2+ \ldots+a_{in}x_n \le b_i \qquad i=1,\ldots,m$$
$$x_1\ge 0, \ldots, x_n\ge 0$$

where f is the objective function, $\mathbf{x}=[x_1, x_2, \ldots,x_n]^T$ is the decision variable vector, $\mathbf{A}=[a_{ij}]_{mxn}$ is the matrix of coefficients of the system of linear inequalities, a coefficient vector

$\mathbf{b}=[b_1, b_2, \ldots, b_m]^T$ will be addressed as a right hand side of the constraints system, $\mathbf{c}=[c_1, c_2, \ldots, c_n]^T$ is a vector of the objective function coefficients.

As we have mention before, in many applications the elements of the tuple (**A,b,c**) cannot be considered as known constants. All or part of them are uncertain. Thus it is impossible to know which solution will appear to be feasible. In such circumstances, one would rather insist on decisions guaranteeing feasibility 'as much as possible'. It is justified by the fact that in such cases constraint violation can almost never be avoided because of unexpected (random) events. On the other hand, after proper estimating the distribution of the random parameters, it makes sense to call decisions feasible (in a stochastic meaning) whenever they are feasible with high probability, i.e., only a low percentage of realizations of the random parameters leads to constraint violation under this given decision. It leads to CCP formulation of the problem, where the deterministic *constraint* are replaced with a *probabilistic* or *chance ones* in the following way:

$$\text{maximize} \quad Ef(x_1, \ldots, x_n) = E(c_1)x_1+E(c_2)x_2+ \ldots+E(c_n)x_n$$

s.t.

$$\Pr(a_{i1}x_1+a_{i2}x_2+ \ldots+a_{in}x_n \le b_i, \ i=1,\ldots,m) \ge q$$
$$x_1 \ge 0, \ldots, x_n \ge 0$$

where $q \in [0,1]$ is the probability level.

The value of probability level is chosen by the decision maker in order to model the safety requirements. Sometimes, the probability level is strictly fixed from the very beginning (e.g., q=0.95,0.99 etc.). In other situations, the decision maker may only have a vague idea of a properly chosen level. It is obvious that higher values of q lead to fewer feasible decisions **x**, and hence to smaller optimal values of expected gain. In some simple cases (especially in case of individual chance constraints) the problem can be replaced with its deterministic equivalent, see e.g. [4], [7].

The main challenge in designing algorithms for general stochastic programming problems arises from the need to calculate conditional expectation and/or probability associated with multi-dimensional random variables. This make the CCP problems most difficult problems of mathematical programming. The computational challenges and methods in the field of optimization under uncertainty are addressed e.g. in [4],[7] and [10]. In our paper we consider the situation where, due to assumed complex stochastic nature of the problem no deterministic equivalent is available. In order to find satisfactory stochastically feasible solution we propose two criteria, one leading to maximization of the probability of feasibility, and the second based on expected utility of a given solution. Then we compare various global optimization algorithms as tools for solving such problems.

III. PROBLEM DESCRIPTION

In our studies we examine the linear programming models in the case where all the parameters determining the problem, i.e. the matrix **A** and vectors **b**, **c**, are random with the expectations equal $E(\mathbf{A})=\mathbf{\Lambda}$, $E(\mathbf{b})=\boldsymbol{\beta}$, $E(\mathbf{c})=\boldsymbol{\chi}$. In the sequel such a problem will be denoted CCLP($\mathbf{\Lambda}$, $\boldsymbol{\beta}$, $\boldsymbol{\chi}$). The performance of the solution found by the Monte Carlo method for the CCLP($\mathbf{\Lambda}$, $\boldsymbol{\beta}$, $\boldsymbol{\chi}$) is compared with the performance of the optimal solution found for the deterministic linear programming problem given by the parameters $\mathbf{\Lambda}$, $\boldsymbol{\beta}$, $\boldsymbol{\chi}$ – in the sequel the latter problem will be denoted by DLP($\mathbf{\Lambda}$, $\boldsymbol{\beta}$, $\boldsymbol{\chi}$).

The decision-maker dealing with the CCLP($\mathbf{\Lambda}$, $\boldsymbol{\beta}$, $\boldsymbol{\chi}$) problem should maximize both the probability q that a given systems of constraints will be satisfied and the expected value of the objective function. However, as it already emphasized, the goals often appear to be contradictory (at least to some extent). Thus in our studies we use two following simulation based indices of performance of a given solution. The first one is the estimated *probability of feasibility* $P_f(\mathbf{x})$ i.e. the probability that the system of constraints will be satisfied when one use a given solution **x**. The second one takes into account both, the estimated probability $P_f(\mathbf{x})$ and the mean value of the criterion function in the case when all constraints are satisfied. It is given by the following formula:

$$SIP(\mathbf{x}) = \frac{N_s}{N_{SIP}} \cdot \frac{1}{N_s} \sum_{i \in S} f_i(\mathbf{x}) \tag{1}$$

where N_{SIP} is a number of i.i.d. Monte Carlo realizations of CCLP($\mathbf{\Lambda}$, $\boldsymbol{\beta}$, $\boldsymbol{\chi}$), N_s is the number of successful realizations (i.e. the realizations for which the system of constraints was satisfied), S is the set of successful realizations indices, $f_i(\mathbf{x})$ is the value of the objective function f obtained in the i-th successful realization of the problem, $i \in S$. A single random realization of CCLP($\mathbf{\Lambda}$, $\boldsymbol{\beta}$, $\boldsymbol{\chi}$) is the realization of a random tuple (**A,b,c**) with the normal probability distribution satisfying the condition $E(\mathbf{A})=\mathbf{\Lambda}$, $E(\mathbf{b})=\boldsymbol{\beta}$, $E(\mathbf{c})=\boldsymbol{\chi}$.

In our simulation study we confine ourselves to problems with positive objective functions.

IV. GRADIENT-FREE GLOBAL OPTIMIZATION ALGORITHMS

"Evolutionary methods" (EM) is a general term that is used to refer to most population-based search methods. EM are population-based metaheuristic optimization algorithms that use biology-inspired mechanisms like mutation, crossover, natural selection in order to refine a set of solution candidates iteratively. So all EM algorithms share the principle of being computer-based approximate representations of natural evolution. These algorithms alter the population solutions over a sequence of generations according to statistical analogues of the processes of evolution. In the literature the EM algorithms are divided into two main groups : *genetic* algorithms and *evolutionary programming methods*.

Genetic algorithms (GAs) are a subclass of evolutionary algorithms where the elements of the search space are binary strings, (sometimes called genotypes). The genotypes are used

in the reproduction operations whereas the values of the objective functions are computed on basis of the phenotypes in the problem space which are obtained via the genotype-phenotype mapping. The algorithm implemented in our simulations can be described as follows, see e.g. [11]:

Step 0 (Initialization) Set the initial population of k vectors $\boldsymbol{x}_i \in R^n$, i=1,2,…,k
chromosomes and evaluate the fitness function for each of the chromosomes.
Step 1 (Parent selection) Select with replacement k parents from the full
population. The parents are selected according to their fitness, with those chromosomes having a higher fitness value being selected more often.
Step 2 (Crossover) For each pair of parents identified in Step 1, perform crossover on the parents at a randomly (uniformly) chosen splice point.
Step 3(Replacement and mutation) Replace the k chromosomes with the current population of descendants from Step 2. Mutate the individual bits with uniform probability
Step 4 (Fitness and end test) Compute the fitness values for the new population of N chromosomes. Terminate the algorithm if the stopping criterion is met; else return to Step 1.
Step 5. Return the so far best generation and the fitness of its elements

While the GAs have traditionally relied on bit coding, the evolutionary programming EP have operated with the more natural floating-point representations. In difference to other types of evolutionary algorithms, in evolutionary programming, a solution candidate is thought of as a species itself. Hence, mutation and selection are the only operators used in EP and recombination is usually not applied., see [13]. In our simulation we use the so-called algorithm of the evolutionary search with soft selection (ES-SS) described e.g in [3], [6]. The algorithm implemented in our simulations is as follows:

Step 0. Set the initial population of k vectors $\boldsymbol{x}_i \in R^n$, i=1,2,…,k (it is so called initial parent population).
Step 1. Assign to each vector $\boldsymbol{x}_i$,$i=1,...,k,$ its fitness i.e. the value of the criterion $F(\boldsymbol{x}_i)$.
Step 2. Select parent $\boldsymbol{v}$ by *soft selection* i.e. with probability proportional to the its *fitness*.
Step 3.Create a descendant $\boldsymbol{w}$ from the chosen parent $\boldsymbol{x}$ by its random mutation: $\boldsymbol{w}=\boldsymbol{x}+\boldsymbol{Z}$, where $\boldsymbol{Z}$ is a random n-dimensional vector with coefficients having expected value equal to zero and given standard deviation σ_z.
Step 4. Repeat steps 2 and 3 for k times to create a new k-element generation of n-dimensional vectors (descendants)
Step 5. Replace the parent population with the descendant population
Step 6. Repeat the second to sixth steps until the stopping criterion is met
Step 7. Return the best element found and its fitness.

In our simulations the initial population in both above algorithms was generated as a population of mutations of the optimal solution of the related DLP(Λ, β, χ) problem.

The third method adopted in our studies is the Simulated Annealing Algorithm (SA).

It is perhaps historically first global optimization method based on stochastic search idea. It was developed by Kirkpatrick in the early 1980s although the main idea was introduced in 1953, by Metropolis as a Monte Carlo method for "calculating the properties of any substance which may be considered as composed of interacting individual molecules", [11]. In metallurgy and material science, annealing is a heat treatment of material with the goal of altering its properties such as hardness. Metal crystals have small defects, dislocations of ions which weaken the overall structure. By heating the metal, the energy of the ions and, thus, their diffusion rate is increased. Then, the dislocations can be destroyed and the structure of the crystal is reformed as the material cools down and approaches its equilibrium state. When annealing metal, the initial temperature must not be too low and the cooling must be done sufficiently slowly so as to avoid the system getting stuck in a meta-stable, non-crystalline, state representing a local minimum of energy.

For the global optimization purpose the idea can be implemented in various ways. The algorithm implemented in our studies is as follows:

Step 0 (Initialization) Set an initial temperature T and initial solution $\mathbf{x} = \mathbf{x}_{curr}$; determine the criterion value $F_C=F(\mathbf{x}_{curr})$
Step 1 Relative to the current value $\mathbf{x}_{curr}$, randomly determine a new value of $\mathbf{x}_{new} \in R^n$, and determine $F_N=F(\mathbf{x}_{new})$
Step 2 Let $d= F_N - F_C$. If $d<0$ accept $\mathbf{x}_{new}$. Alternatively, accept $\mathbf{x}_{new}$ only if a random variable U having the uniform p.d. on the interval [0,1] satisfies $U < \exp[-d/T]$. If $\mathbf{x}_{new}$ is accepted then $\mathbf{x}_{curr}$ is replaced by $\mathbf{x}_{new}$; else $\mathbf{x}_{curr}$ remains as is.
Step 3 Repeat steps 1 and 2 for given number k times.
Step 4 Lower T according to the annealing schedule and return to Step 1. Continue the process until the stopping criterion is met.
Step 5. Return the best solution $\boldsymbol{x}_b$ found during the cooling process and the value $-F(\boldsymbol{x}_b)$

The specifics of implementation for the steps above can vary greatly. In our simulation the initial solution $\mathbf{x} = \mathbf{x}_{curr}$ in Step 1 is generated as a mutations of the optimal solution of the DLP(Λ, β, χ) problem. The initial "temperature" T decays geometrically in the number of cooling phases (number of times T is lowered according to step 4). Specifically, the new temperature is related to the old temperature according to $T_{new} = 0.75\ T_{old}$. Another area for different implementations is in step 1, where $\mathbf{x}_{new}$ is generated randomly. In our studies it was generated according the formula $\mathbf{x}_{new}=\mathbf{x}_{curr}+\boldsymbol{Z}$, where $\boldsymbol{Z}$ is a random n-dimensional vector with coefficients having expected value equal to zero and given standard deviation σ_z. Also note that the simulated annealing algorithm is design for minimization tasks, thus the criterion used in the algorithm is given by opposite values to the original ones which is reflected in the last step.

For any given problem, it is likely that the performance of one algorithm will be superior to others. A priori, of course, it is rarely possible to know which algorithm is superior. Famous No Free Lunch theorems state that *when averaging over all*

optimization problems all search algorithms work the same (i.e., none can work better than a blind random search), [11]. However the NFL theorems do not address the performance of a specific algorithm applied to a specific criterion function. They compare the performance of algorithms over all problems, where each problem is considered equally likely. On the other hand it is well known that for some specific types of problems some algorithms may perform extremely well, while other perform very poorly. Our aim is to determine which of the above algorithms (if any) is best for solving the stochastic linear programming problems.

V. SIMULATION STUDY OF THE ALGORITHMS

To compare the stochastic performance of various solutions **x** we use the performance indicators $P_f(\mathrm{x})$ and $SIP(\mathrm{x})$. The obtained values will be compared also with the characteristics of the optimal solution obtained in the related deterministic problem. For this purpose we use the rate between the $SIP(\mathbf{x})$ and the optimal objective function value $\max_D$ in deterministic case, i.e. the indicator $SDR = \dfrac{SIP(\mathbf{x})}{\max_D}$

The values of the indicator obviously depend on the problem parameters, i.e. on the tuple $(\mathbf{\Lambda}, \mathbf{\beta}, \mathbf{\chi})$ and the dimensions of its elements. Thus we compute the values of the indicators for various values of the parameters and compare its statistical characteristics: maximum value, minimum value, mean value and standard deviation. In order to obtain the statistical data we use the following simulation procedure.

Step 0. Set the parameters n , k, N_{SIP}, N_G and K.
Step 1. Randomly generate the tuple $(\mathbf{\Lambda}, \mathbf{\beta}, \mathbf{\chi})$
Step 2. Solve the DLP$(\mathbf{\Lambda}, \mathbf{\beta}, \mathbf{\chi})$ problem by the simplex algorithm and obtain the solution $\mathbf{x}_D$, the optimal value $\max_D = f(\mathbf{x}_D)$ and $SIP(\mathbf{x}_D)$
Step 3. Solve CCLP$(\mathbf{\Lambda}, \mathbf{\beta}, \mathbf{\chi})$ problem by the given GO algorithms, obtain the solutions and the related values of the criterion functions ($P_f(\mathbf{x})$ or $SIP(\mathbf{x})$, respectively)
Step 4. Compute and record the values of the indices SDR for this setup
Step 5. Repeat the first to forth steps for K times, where K is a sufficiently large number.
Step 6. Return: statistical characteristics of the results such as maximum , minimum , mean values and standard deviations of $P_f()$ and SDR.

In our research we use the following values of the parameters: N_{SIP}=500, n=4,8,12 and k=10, K=30. In the ES-SS algorithm the distributions of mutations are normal with constant standard deviation equal to 0.1. The values of the parameter m are drawn from the set $\{n\text{-}2,...,n\text{+}5\}$. The elements of the tuple $(\mathbf{\Lambda}, \mathbf{\beta}, \mathbf{\chi})$ are drawn from the interval [-200,700]. In step 1 the distributions of each element of the matrix and both vectors are normal with mean equal zero and standard deviation being equal to 10% of the value of the appropriate element in the deterministic case.

One of the key issues in comparing algorithms is "efficiency" in solving problems of interest. The efficiency measure is some representation of the cost of finding an acceptable solution. There are many ways of measuring efficiency: computer run time, number of algorithm iterations, and number of the criterion function evaluations. In our paper we use the latter as generally the most objective measure, see e.g.[11]. A serious motivation is that in practice the criterion measurements are usually the dominant cost in the optimization process; especially in the case of measurements based on Monte Carlo simulations. In our simulations each criterion function value measurement requires 1000 generations of the random setup of the problem.

VI. RESULTS AND FINAL REMARKS

In the following tables we present the results of our simulations. The results were obtained in simulations in which all algorithms stop working after 300 measurements of a given criterion function (for each random setup of the problem).

Table I shows the statistical characteristics of the probability $P_f()$ computed for the best solution found with the help of an indicated algorithm as well as for the optimal solution found in the related deterministic problem (DP). The dimensions of the simulated problems are n=4,8,12.

TABLE I. STATISTICAL CHARACTERISTICS OF THE PROBABILITY $P_f()$ COMPUTED FOR THE BEST SOLUTIONS

Algorithm	Min	Max	Mean	St.Dev.
	n=4			
ES-SS	0.672	1	0.992	0.046
GA	0.16	1	0.850	0.261
SA	0.048	1	0.908	0.263
DP	0.016	1	0.201	0.169
	n=8			
ES-SS	0.564	1	0.966	0.101
GA	0.076	1	0.847	0.288
SA	0.004	1	0.759	0.400
DP	0.004	0.452	0.089	0.096
	n=12			
ES-SS	0.136	1	0.872	0.211
GA	0.06	1	0.548	0.383
SA	0	1	0.597	0.441
DP	0	0.22	0.264	0.044

We see in the Table I that all statistical characteristics of the results obtained for various dimensions of the problem show the dominance of the ES-SS algorithm with respect to this criterion - so it is the most effective algorithm in finding statistically feasible solutions (we remember that the parameter $P_f()$ indicates the probability of feasibility) .

Next table presents analogous results, but this time connected with the criterion function being the indicator SDR.

TABLE II. STATISTICAL CHARACTERISTICS OF THE PROBABILITY *SDR* COMPUTED FOR THE BEST SOLUTIONS

Algorithm	Min	Max	Mean	St.Dev.
	n=4			
ES-SS	0.381	1	0.759	0.148
GA	0.166	1	0.627	0.250
SA	0.005	0.932	0.698	0.233
DP	0.024	1	0.223	0.214
	n=8			
ES-SS	0.459	0.914	0.746	0.107
GA	0.074	0.946	0.595	0.242
SA	0.020	0.934	0.630	0.226
DP	0.004	0.252	0.082	0.069
	n=12			
ES-SS	0.105	0.887	0.660	0.163
GA	0.014	0.919	0.468	0.256
SA	0.004	0.907	0.438	0.315
DP	0	0.176	0.036	0.042

Again, similarly as in the previous table, one can notice the dominance of the ES-SS algorithm. However this time we see in the column containing the maximum values of SDR, that in some problems generated in our simulations other algorithms (GA or SA) perform better than ES-SS. On the other hand all other characteristics indicate that the ES-SS algorithm is the most suitable tool for solving chance constrained programming problems, at least in the situations considered in this paper.

Finally, let us emphasize again that the solution for CCLP(Λ, β, χ) problem found by the evolutionary search may be called satisfactory rather than optimal. The optimality cannot be proved - it is a typical situation when one use the global optimization algorithms based on stochastic search. But the solution is relatively easy to find and, in considered stochastic framework, much better than the optimal solution found in deterministic case. The solution may be considered as satisfactory especially because of the high values of the indicator *SDR*. Taking into account that the standard deviations of random variables disturbing all elements of the tuple (Λ, β, χ) amounts to 10% of its original values, one should not expect much more, even when applying (where it is possible) more sophisticated mathematical tools.

REFERENCES

[1] A. Charnes, W.W. Cooper, "Chance-constrained programming", Management Sciences 6, (1959),73–80.

[2] P.K.De , D. Acharya, K.C. Sahu, "A chance-constrained goal programming model for capital budgetting", Journal of the Operational Research Society 33 (7) , 1982, pp. 635–638.

[3] R. Galar, Soft selection in random global adaptation in Rn: A biocybernetic model of development, Technical University Press, Wrocław, 1990, (in Polish).

[4] P. Kall, J. Mayer, Stochastic Linear Programming; Models, Theory, and Computation, Springer, New York London, 2011.

[5] V.I. Norkin, G.C. Pug, A. Ruszczyński, "A branch and bound method for stochastic global optimization", Mathematical Programming: Series A and B Volume 83 , Issue 3, 1998, pp. 425 – 450.

[6] A. Obuchowicz, J. Korbicz, "Global optimization via evolutionary search with soft selection", unpublished manuscript, available: http://www.mat.univie.ac.at/~neum/glopt/mss/gloptpapers.html

[7] A. Ruszczynski, A. Shapiro, Stochastic Programming. Handbooks in Operations Research and Management Science, Vol. 10. Elsevier, Amsterdam, 2003.

[8] G.I. Schuëller, H.A. Jensen, "Computational methods in optimization considering uncertainties – An overview", Comput. Methods Appl. Mech. Engrg. 198,2008, pp. 2–13.

[9] S. Sen, J.L. Higle, "An introductory tutorial on stochastic linear programming models", Interfaces, 29, 1999, pp. 33-61

[10] S. Sen, "Stochastic programming: computational issues and challenges", in: Encyclopedia of Operations Research and Management Science, 2nd edition, ed. S. Gass and C. Harris, 821-827. Boston: Kluwer Academic Publishers, 2001.

[11] J.C. Spall, Introduction To Stochastic Search And Optimization; Estimation, Simulation, And Control , A John Wiley & Sons. Inc., Publication, 2003

[12] R. Steuer P. Na, "Multiple criteria decision making combined with finance: A categorized bibliographic study", European Journal of Operational Research 150, 2003, pp. 496–515.

[13] T. Weise, Global Optimization Algorithms – Theory and Application, available http://www.it-weise.de/projects/book.pdf (accessed 7.09.2011)

Simulation Environments for Processes Control Verification Using Hardware in Simulation's Loop

Mikuláš Alexík
dept. of Technical Cybernetics, University of Žilina,
Žilina, Slovak Republic
Mikulas.alexik@fri.uniza.sk

Abstract— **This paper describes method for realisation of configurable time continuous linear and non-linear models based on microcontroller (Atmel ARM7). Its can be used for verification of identification algorithms, transport processes dynamics and behaviour of control loop using simulation experiments, if simulations runs in real time. Application of microcontroller enable us simple realisation of changing parameters and non-linearity's in the process. In the paper will be attention paid to realisation of "real time" in programmable environment "Windows" and realisation of precise calculations in floating point on microcontroller's. (Abstract)**

Keywords-component; microcontroller; real time; A/D and D/A converters; time delay (key words)

I. INTRODUCTION

Based on former experience from research and teaching [3], [4], [5] [6] we can confirm, that laboratory verification of all algorithms with processes dynamics using simulation experiments, presents verifiable results just if simulation experiments are executed in real time. Such a manner of simulation experiments execution means, that inside the simulation loop runs dynamical process in time independent from the simulation software. Then simulation software has to respect change of dynamics and variables in the independent process, in the other words, the simulation software (its speed) has to be adapt to the speed and precision of the real physical process.

The most used method is applying the special laboratory equipments that are physically realised models of real systems as helicopter model or inverse pendulum. These laboratory models together with A/D and D/A converters, which for us represents "hardware in the loop of simulation" are connected with PC where is control algorithms programmed plus some software program which enable us to realise verification using simulation experiments. This methods is very visual advertising in teaching process but is not very applicable in research, because there is no way to change process parameters and non-linearity, especially time delays.

Application of microcontroller as the base of model, enable us simple realisation of changing parameters and non-linearity's and especially transport delays in the process [1]. In the paper will be attention paid to realisation of "real time" in programmable environment "Windows" and realisation of precise floating-point calculations on microcontroller.

The paper is organized as follows. Section 2 describes proper models of the system dynamics. Section 3 describes real time response models of the processes. Section 4 describes real time simulation experiments with PC and hybrid simulation. The paper ends with conclusion and outlook in Section 5

II. MODELS OF THE SYSTEM DYNAMICS

For verification of dynamics of "driver in the car" authors of this article verified [3],[5] models (1) For verification of control algorithm, authors of [2] recommend to carry out verification with transform functions (2), which cover wide-spread of technological systems behaviour.

$$S_1(s)=\frac{K\left(1+T_3*s\right)*e^{-d*s}}{\left(1+T_1*s\right)\left(1+T_2*s\right)},\quad T_1,T_2,T_3=0.05\div 0.9$$
$$S_2(s)=\frac{K\left(1+T_3*s\right)*e^{-d*s}}{\left(T_1^2*s^2+2T_1*a*s+1\right)},\quad a,d=0.05\div 0.95; \tag{1}$$

$$S_1(s)=\frac{e^{-\alpha*s}}{\left(1+T*s\right)^2},\quad T=0,1,...10;\ \alpha=0.1...10$$
$$S_2(s)=\frac{1}{\left(1+s\right)^n},\quad n=3,4,...8$$
$$S_3(s)=\frac{1}{\left(1+s\right)\left(1+\alpha*s\right)\left(1+\alpha^2*s\right)\left(1+\alpha^3*s\right)},\alpha=2\div.7 \tag{2}$$
$$S_4(s)=\frac{\left(1-\alpha*s\right)}{\left(1+s\right)^3},\quad \alpha=0.1,0.2,0.5,1,2$$

Both types of models described dynamics with type of non-linearity "d,α" named "transport delay". In the analytical mathematics this models are represents by non-linear differentials equations. Its can be modeled in real time by continuous electronics environments (HIL simulation). More simply is the "PC simulation" when the output of the models is calculated (not in real time) as "difference equation" on the computer. Environment described in this paper enables to realize all sorts of non-linear dynamics. The realization possibilities are in detail describes in next parts of the paper.

III. Real Time Response Models of the Processes

Transfer function (1), (2) can be realized by next mode:

a) Analogue model of the processes realized by analogue hardware. This mode has the problem with realization of time delay. This possibility was realized and verified by author. More detail is in [1].
b) Analogue model of the processes realized by FPAA – Field Programmable Analogue Array. This mode was not verified by author but was described in [1].
c) Hybrid model of the processes can be realized by digital computation in real time on PC and through A/D and D/A convert as continuous input/output to the/from the process. The control algorithm is computed in real time and A/D and D/A converters realize connection to plant. This mode is describes in this section, part *A*.
d.) Hardware in loop model of the processes, realized by any microcontroller + A/D and D/A converters. This mode is describes in this section, part *B*.

A. Hybrid Environment with Computation on PC

This mode is a digital computation of controlled plant dynamics and control algorithms behaviour on PC in real time, combined with realization of input/output for plant and controller with A/D and D/A converters. This means that in comparison to measurement mode, twice as much converter is needed, because output/input to the controlled plant have to be in analogue mode. Mode block scheme is depicted on Figure 1.

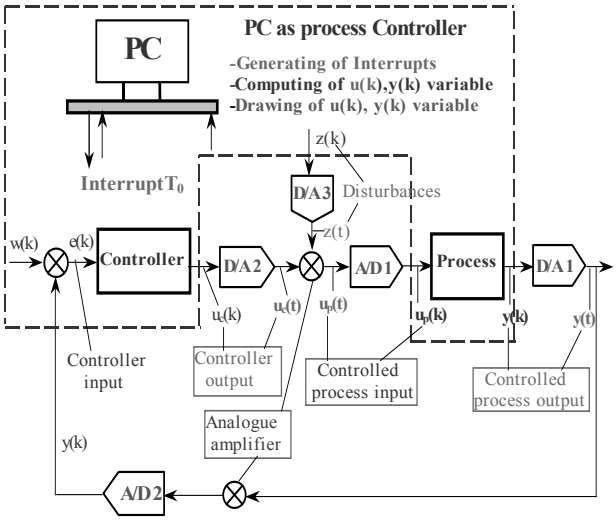


Figure 1. Real time simulation of controller and plant in the same PC

On the block scheme it can be seen two analogues adding amplifier with D/A as input and A/D as output. Applying of D/A3 for "disturbance" input is more suitable as mode where "disturbance" is as analogue signal. It is simply generate in analogue mode disturbance signal "unit step" but is impossible generate in this mode pseudorandom signal, which is no problem to generate from PC as input to the D/A3 converter. For both modes applying of analogue amplifier it is necessary for summation of two analogue signals before A/D. The block scheme are be used also without of second amplifier. If the analogue amplifier is good realized, then can reduce "quantization error" (Figure 2, error for 12-bit A/D) which is for 12-bit D/A converter bigger than for 14-bit A/D converter.

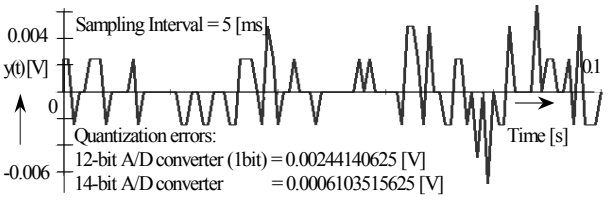


Figure 2. Quantisation error for on analogue amplifier output

B. Hybrid Models Realised on Micro Controller

When verifying analog model control working in real time, it is needed to apply A/D and D/A converters. Computations needed for control algorithm are realized as parallel in real time as well. In this „hardware in loop simulation" (HIL), while measuring is done with A/D converters, there is an unpredictable signal noise, different in every simulation experiment. The same qualitative situation occurs in real process control. That is why the verifying quality in HIL control algorithm is higher than in digital simulation. The differences between model and real process, including quantization signal noise, influence to the quality of regulation process quite a lot. It can be negative influenced as well by delay while computing control algorithm. Therefore it is useful for processes with delay (especially for control through internet environment) to have algorithms verified on analog models in real time. The scheme with connecting slots of hybrid model is pictured on Figure 3. On Figure 4 is microcontroller part of model and on Figure 5 is part of model block scheme.

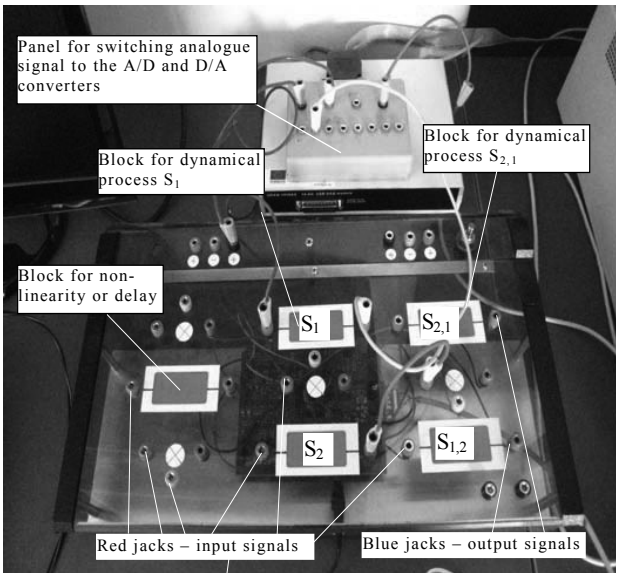


Figure 3. Structure of Hybride model based of microcontroller

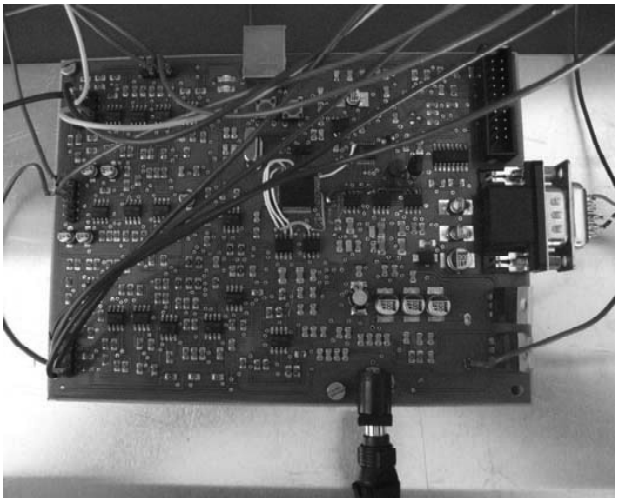


Figure 4. Microcontroller part of hybrid model

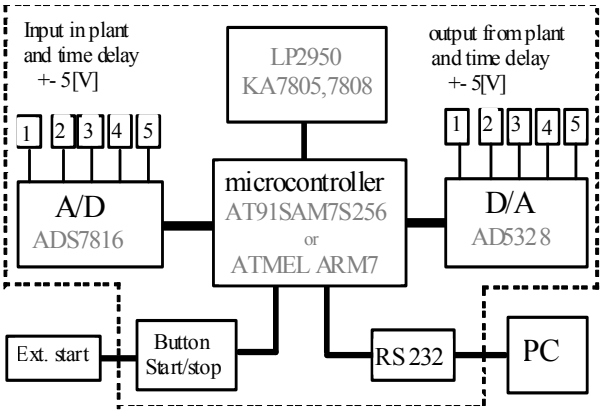


Figure 5. Microcontroller part block scheme

Connection the microcontroller with PC through RS 232 is used for sending of Z-transformation coefficient computed in PC to microcontroller. USB connection is using in time of microcontroller programming.

To verify the hybrid model, simulation experiments with controlled plants and various controlled loop were carried out. Controlled processes dynamics were compared with batch processes against [1], detailed description is in [3], [5], [7]. Examples of simulation experiments with hybrid model in control loop are commented in the next part of this paper. For microcontroller AT91SAM7S256, only 32 bit arithmetic is available. The coefficients for controlled process model in Z-transformation are computed in PC with 64 bit floating point accurately (relative – it depends from sampling interval). Rewritings the coefficients to microcontroller where is only 32 bit arithmetic causes many problems in precision of calculation for difference equation output. This problem is explained in the next part of paper.

C. *Calculation of dynamic processes responses*

Dynamic response of used model is computed from PC independently in the microcontroller, as output from difference equation (4), which is rewriting of Z-transform function (3). (In next consideration it is suppose that delay $d=0$, which have no influence on results about precision of calculations of Z-transform). This Z-transform function is computed in the PC for exactly and uniquely determined sampling interval T_{0y}. The outgoing values from the model of dynamical process have to be identically for computation in analogue mode (1) or discrete mode (3), for identical input signal. As can be seen from (4) the situation in calculation from digital mode strictly depended from precision of calculation of parameters "a_i, b_i" for Z-transform function. And its precision depends from used floating point and also, as can be seen in (3), from used "sampling interval". For clear example we suppose situation only in steady state mode. If the input signal is unit step, then for dynamic process described by (1), steady state has to be gain K for analogue and digital mode, but also for calculation from transform function (1) or (3), which is documented in (5).

$$S(z)=\frac{b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}=\frac{Y(z)}{U(z)};\quad a_2=D_1D_2;\quad D_i=e^{-\frac{T_i}{T_{0y}}}$$
$$b_2=K\left(D_1D_2-D_2\frac{T_1-T_3}{T_1-T_2}+D_1\frac{T_2-T_3}{T_1-T}\right) \tag{3}$$

$$y(k)=b_1u(k\text{-}1)+b_2u(k\text{-}2)-a_1y(k\text{-}1)-a_2y(k\text{-}2)$$
$$\text{Shift } u(k)\equiv u(k\text{-}2)=u(k\text{-}1);\ u(k\text{-}1)=u(k);\ \text{Shift } y(k) \tag{4}$$

$$\text{if } u(t)=\text{unit step}=1(t)\ \Rightarrow U(s)=(1/s)$$
$$\lim_{t\to\infty}\left[y(t)\right]=K=\lim_{s\to0}\left[sY(s)\right]=\lim_{s\to0}\left[sS_1(s)U(s)\right]=$$
$$=\lim_{s\to0}\left[sS_1(s)\frac{1}{s}\right]=\lim_{s\to0}\left[S_1(s)\right]=K \tag{5}$$
$$\lim_{t\to\infty}\left[y(t)\right]=\lim_{k\to\infty}\left[y(k)\right]=\lim_{s\to0}\left[S_1(s)\right]=\lim_{z\to1}\left[Z_1(s)\right]\overset{!}{=}K$$
$$\lim_{z\to1}\left[Z_1(z)\right]=\lim_{z\to1}\left[\frac{b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}\right]=\frac{b_1+b_2}{1+a_1+a_1}\overset{!}{=}K$$

Situation "around precision" is clear documented in (6), where results of parameters a_i, b_i and gain K calculation in 32-bit and 64-bit floating point are figured. There are figured also parameters, which are needs to use for the problem solving.

After the parameters in 64-bit arithmetic, which were calculated on PC, and before than parameters will be sent to the microcontroller, it is necessary to shift parameters by 6-position left (multiplication with E6). After calculation the output from difference equation (4) and before the output will be drawing or typing (and shifting in (4)), it is needed to shift output value by 6-position right (multiplications with E-6). This solution was designed and verified by authors.

On the Figure 6 step responses for transfer function from example (6) are figured. It is clearly documented that computation of dynamics by difference equations with 32-bit arithmetic unsecured right calculation.

For $K = 1;\ T^2 = 4;\ 2aT = 0,1;$ Sampling=1 [ms]

Parameters of Z-transform in 64-bit mode are:

$b_1 = 1.24998955766\,E^{-7}$; $b_2 = 1.249979140575\,E^{-7}$;

$a_1 = -1.9999747503156$; $a_2 = 0.99997500031249$;

Gain $K = 1.000$;

(6)

In 32-bit mode: Gain $K = 0.83885$

$b_1 = 1.2499896229\,E^{-7}$; $b_2 = 1.2499791069\,E^{-7}$;

$a_1 = -1.9999747276$; $a_2 = 0.99997502565$;

32-bit mode after shifting E6; Gain $K = 0.999987$

$b_1 = 0.12499895692$; $b_2 = 0.12499791384$;

$a_1 = -1999974.75$; $a_2 = 999975.0$;

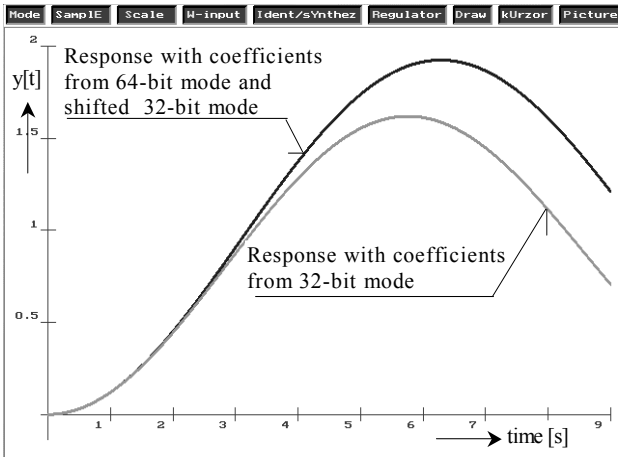


Figure 6. Step response for 64-bit and 32-bit floating point arithmetic

Because computation on microcontroller is independently from PC, and control algorithms is computed on PC, it is necessity to ensure synchronization between PC and microcontroller. This problem is described in next part *D*.

D. Realization of Real Time under environmend Windows

Realization of real time is the main problem in the hybrid and HIL simulation mode. This paper is focused on problem of dynamic processes realization by difference equation computed in microcontroller with 32-bit floating point arithmetic. Only list of main mode is declared without comment therefore.

1. MS DOS environment + interrupt service routine controlled by 8253 timer.

2. Windows 98 + MS DOS + interrupt, or Windows CE.

3. Special edition of Linux environment

4. Windows 2000, XP, 7 + multimedia timer in environments Borland C, Borland Delphi

5 Windows, after 98 + RTX environment.

6. Windows after 98 + ZIL timer.

Author of this paper has experiences with first 5 modes. Most often he used the second mode, because in this mode is direct time interrupt. In the first three modes the handler for A/D and D/A converters is being executed in time interrupt and it makes possible to check duration of control algorithms computation. From the real time simulation experiment duration point of view (3 – 150 [s]) the sampling interval 1– 50 [ms] was being used. The top time 50[ms] it is fixed by simple realization of interrupt service program and hardware realization of time register in 8253 timer.

The application of multimedia timer unsecured strict sampling interval realization. Its precision can be improved by suitable realization of application program and optimalization of OS Windows environment (special environment with only needed service programs of OS, without all services for network). From author experiences point of view, this mode can be used, for good organized application program, with 10 [ms] sampling interval, even for not modified OS Windows .

Application of real time mode with RTX (Real Time Extension) environment it is conditioned by multicore processors applying. One of the processors core it is reserved for RTX and interrupted input/output handling. Usually was RTX applied together with serial RS 232 port as input/ouput to the PC. Disadvantage of this mode is that cannot by used for interrupting of USB port. RTX environment is not free software and is quite complicate to using.

ZylTimer is a high resolution, long-term Delphi and C++ Builder timer component which provide a higher precision than the standard Delphi/C++ Builder TTimer component. ZylTimer is a thread based timer and due to this architecture provides a higher precision, close to 1 millisecond (it's possible that you cannot obtain a clear resolution of 1ms on all the systems, but it must be under 2ms) which is inevitable in time critical applications.

IV. REAL THE SIMUALTION EXPERIMENTS

Real time simulation experiments were realized with analogue model of controlled plant [1] or hybrid mode described in the previous section. Real time continuous identification of plant parameters consecutiveness of control algorithm parameters synthesis enabled us to realize adaptive control with algorithm described in [7], [8]. By using this approach the controller is updated at every sampling interval and the incoming input-output information is capable to improve the parameter estimation process. In principle, the on-line mathematical model of controlled plant and on-line synthesis of control algorithm for demanded quality of control process were suggested. This advance is called Self-Tuning Control (STC) or indirect adaptive control. Example of real time simulation experiment with continuous identification and control from aptitude of proportional third order controlled plant model apply for control loops with higher order of plant. One of examples is focused to the realisation and control of processes with time delay. As were commented before this type of processes cannot be realised by analogue environment, therefore hybrid mode were made-up for its realisation.

Comparison of computer simulation and hybrid simulation of adaptive PIDD^2 control, including identified parameter

tuning by hybrid simulation, is shown in Figure 6 and Figure 7. Controller synthesis was realized for third order-identified model. Synthesis of continuous and discrete PIDD^{2}algorithm from identified data, derived by author is described in [5]

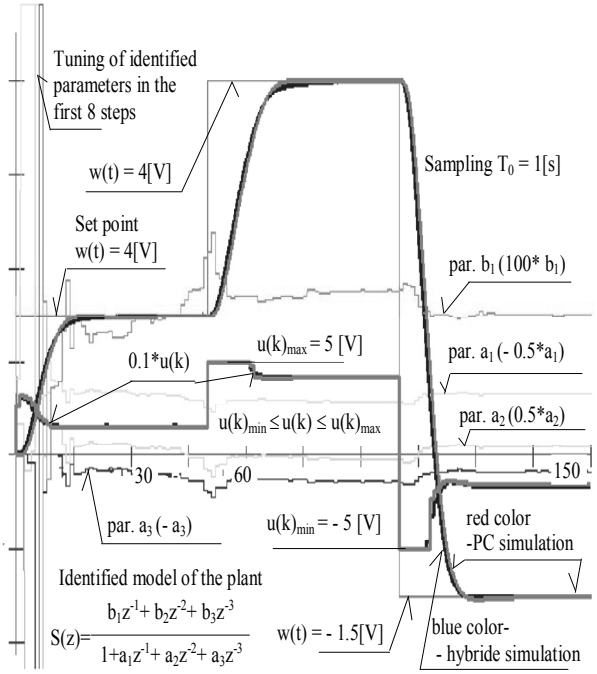


Figure 7. Comparison of PC and hybride simulation

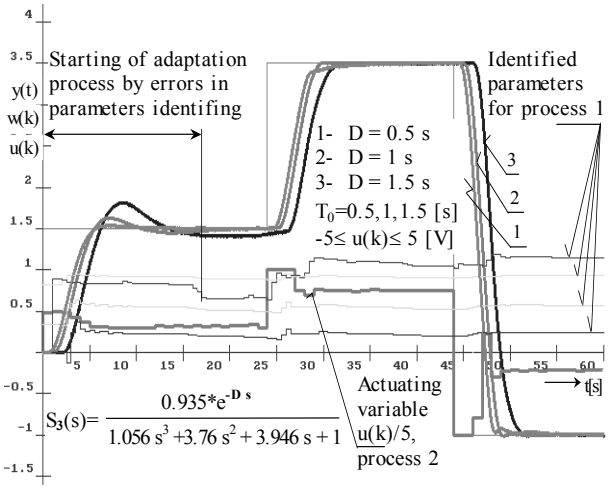


Figure 8. Control of classical time delayed process – PC simulation

The simulation experiment shown at Figure 8 is hybrid simulation of adaptive control of plant with dead time for different size of sampling interval. To control such type of plant is the most difficult task, but from Figure 8 and Figure 9 it is evident, that eDB-d [9] adaptive algorithm provides very good quality of controlled loop response. At present time author can to identified dead time for one to three sampling interval and also parameters of third order process. The verification of this type of algorithms in real time can be prepared only after realization of hybride or HIL simulation experiments mode.

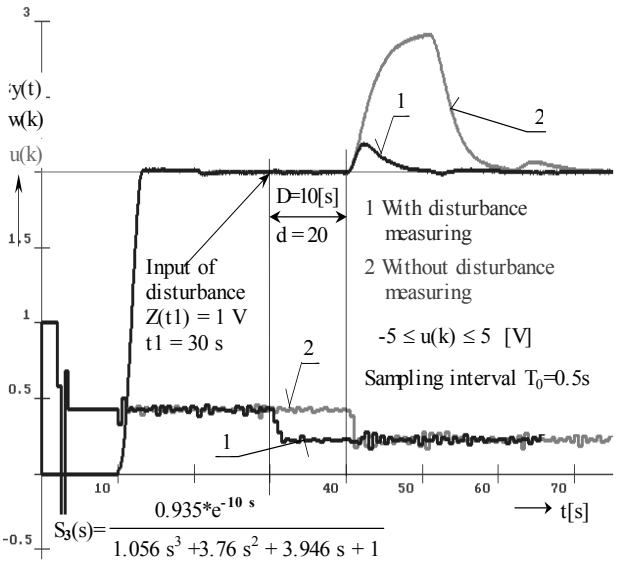


Figure 9. Control of process with dominating time delay- hybride simulation

Real time HIL simulation of a closed loop system with process whose dynamics is dominated by dead time shows Fig. 9. The Figure shows that the load-disturbance response is excellent for case with disturbance measuring however in common is dominant opinion that dead time produce load-disturbance response with big overshoot and settling time.

The fact, that simulation experiments were realized in real time mode with A/D and D/A converters can be seen on the Figures by time responses of controlled variable (y(t)) and control variable (u(k)). Both variables especially in steady state mode are chalk-line in PC mode (red color line on Figure 7). But in the hybrid mode A/D and D/A converter generate quantization error, which is clear seen from Figures 8, 9 on control variable u(k) and controlled variable y(t). In the all real time simulation experiments the sampling interval for controlled variable y(k) is 10 – 50 times smaller than sampling interval for u(k). Therefore frequency of noise on y(k) is larger than for u(k). The presence of noise in documentation of simulation experiments is suitable character for recognition if simulation experiment was prepared only as PC simulation or hybrid simulation.

V. CONCLUSION

Based on present experience, the laboratory verification and teaching of control algorithms in described HIL simulation provide good results. This way, classical and adaptive PID, PIDD^2, DB(n), eDB, eDB-d, algorithms, state algorithms, sliding mode algorithms were verified. This paper described

two simulation environments for control algorithms verification and other dynamic processes realization. Simulation experiments with digital models of controlled systems and real time HIL simulation were realised in program environment of ADAPTLAB, developed and realised by author. This environment is suitable for developing and verification of classical as well as adaptive control algorithms for SISO and MIMO control loops. The program can be used in three basic modes: Simulation, measurement and hybrid mode. The simulation mode works with continuous transfer function set by operator. In the measurement mode the output (input) from model of the plant realised on microcontroller (part III B), is measured with A/D (D/A) converter with sampling interval controlled by real time clock from PC or from A/D converters. Hybrid mode (part III A) is a digital computation of controlled plant behaviour and control algorithms on PC in real time, combined with realization of input/output for plant and controller with A/D (D/A) converters. A part of described environment is successfully used in laboratory practically in subjects "Digital Control", "Computer Control" and "Simulation of the Systems" at the University of Zilina. The environment also helps developing and verification of new adaptive control algorithms. Up to now experience of author showed, that HIL simulation is more appropriate as digital simulation both for verification and teaching.

In future, the author will focus on some self-tuning algorithms problem and variable transport delay identification. In area of "intelligent transport systems" the research will focus on the measurement, identification and modeling of dynamics in the environment "driver/car traffic situation"

ACKNOWLEDGMENT

This contribution is prepared under the projects VEGA 1/1099/11 "Modeling and Simulation of Dynamical Interaction in Environment of Driver/Car/Traffic Situation ".

REFERENCES

[1]] Alexík S., Alexík M.: Simulation Environment for Real Time Verification of Control Algorithm. In: Proceedings of 5th Vienna Symposium on *Mathematical Modelling*. February 8-10 2006, Vienna University of Technology, Austria. ARGESIM Report no.30. pp. 3-1 – 3-10. Published by ARGESIM and ASIM, Arbeitgemeischaft Simulation Fachausschuss GI im Bereicht ITTN –Informationstechnik und Technische Nutzung der Informatik. Volume 2: Full Papers CD. ISBN 3-901608-30-3

[2] Astrom, K. J. and T. Häglund: *PID Controllers, 2nd Edition.* Instrument Society of Amerika, Research Triangle Park, North Carolina.,1995.

[3] Alexík, M: Modelling and identification of eye-hand dynamics. Simulation Practice and Theory, Vol 8, 2000, pp.25-38.

[4] Alexík, M.: Modification of Dead Beat Algorithm for Control Processes with Time Delay. Proceedings of 16th IFAC WORLD CONGRESS, Prague, July 4-8. 2005. paper n. 3402.

[5] Alexík, M.: Modelling and Simulation of Interaction in Driver/Vehicle Dynamics. Proceedings of 6th Vienna Conference on Mathematical Modelling (MATHMOD'09), February 11-13, 2009 Vienna University of Technology, Published by ARGESIM and ASIM, ARGESIM Report no.35, Full Papers CD Volume. pp. 500-510. ISBN 978-3-901608-35-

[6] Alexík M.: Modelling of Process Control that have Time Delay. In: Proceedings of 12th International Conference on Modelling and Simulation (UKSIM 2010), 24-26 March, 2010 Cambridge University (Emmanuel Colledge), United Kingdoom. Published: IEEE Computer Society Conference Publishing Service (CPS). IEEE Service Center, 222 Rosewood Drive, Danvers, MA 01923. DOI 10.1109/UKSIM.2010.48; pp 221-226. ISBN-13: 978-0-7695-4016-0

[7] Alexík, M: Usage of Real Time Hybrid Simulation for Verification of Control Algorithms In: Proceedings of the 4th International Eurosim 2001 Congress "SHAPING FUTURE WITH SIMULATION", TU Delft, The Netherlands, 26-29 June 2001, paper. number. 052, Delft University of Technology. ISBN 90-806441-1-0.

[8] Alexík M.: Simulation Experiments with Self Tuning PSD Control Algorithm. In: Proceedings of UKSim Tenth International Conference on Modelling and Simulation, EUROSIM/UKSim2008, 1-3 April 2008, Cambridge, Uk (Emmanuel Colledge). Copiright © 2008 by IEEE Inc., pp. 34-39. Product Number E3114. Editorial and CD-ROM production by Stephanie Kawada, ISBN 0-7695-3114-8. LoC Number 2007941927.

Software Package "FSO System Simulator": Design and Analysis of the Steady State Model for the FSO Systems

Pavol Mišenčík, Matúš Tatarko
Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics
University of Technology Košice
Košice, Slovakia
pavol.misencik@tuke.sk; matus.tatarko@tuke.sk

Ľuboš Ovseník, Ján Turán
Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics
University of Technology Košice
Košice Slovakia
lubos.ovsenik@tuke.sk; jan.turan@tuke.sk

Abstract— **This paper describes a software package FSO System Simulator (FSO SystSim) which was designed and implemented at KEMT FEI TUKE. Simulation of FSO communication link is of great importance in designing and understanding the context of such connection depending on various parameters (technical and constantly changing atmospheric parameters of the transmission optical channel). Paper briefly describes the static model used in this programming package and describes experiments carried out by the FSO SystSim.**

Keywords-FSO simulator; modeling; software package

I. INTRODUCTION (HEADING 1)

The atmospheric optical links in visible and infrared wavelength constitute an interesting alternative for a variety of applications in telecommunications field [1,2,6,7,9-11]. Several factors determine the revival of such technique: free licence, easy, fast and inexpensive deployment and high data rates.

Before the implementation of effective FSO communication links we need to know their availability and their reliability (percentage of time during which a value is reached or exceeded). Availability and reliability of FSO communication link depends on used systems but also on atmospheric parameters such as rain, snow or fog [4,8,9-11]. This is the purpose of our study. Its output is a software FSO SystSim which with input parameters (distance, Tx power, Rx sensitivity, Rx lens diameter, directivity of laser, weather conditions etc.) allows to determine the availability of a communication link. Detailed description of the program is shown in the following sections of this article.

II. STEADY MODEL

As already mentioned, FSO optical links have to operate in changing conditions that cannot be accurately predicted or avoided – changing weather. It is therefore necessary to know the behavior of the line under any conditions and be able to choose appropriate parameters for communication in particular environment.

Whether optical link will operate on 100 meters or 1000 meters distance (with respect to the parameters for receiving and transmitting devices such as transmitted power, the diameter of the lens receivers, receiver sensitivity, directivity laser beam, the distance between the transmitter and receiver) is in charge of steady model of the optical communication link. The basis for this model is power budget model.

A. Power Budged Model

The basic formula for calculating the balance of power of FSO link is very simple [3]. Values in dB can be expressed as:

$$P_{m,RXA} = P_{m,TXA} - \alpha_{12} - \widetilde{\alpha}_{atm} - \alpha_{atm} + \gamma_{tot} \qquad (1)$$

where $P_{m,RXA}$ is the mean of received power, $P_{m,TXA}$ is the mean of emitted power, α_{12} is the attenuation of spread, $\widetilde{\alpha}_{atm}$ is the attenuation of pure atmosphere, α_{atm} is an additional attenuation due to deteriorated weather conditions and γ_{tot} is the gain of the receiver [3]. For a better understanding of power and loss ratios on the transmission path, power diagram of FSO link is shown on *Fig. 1*.

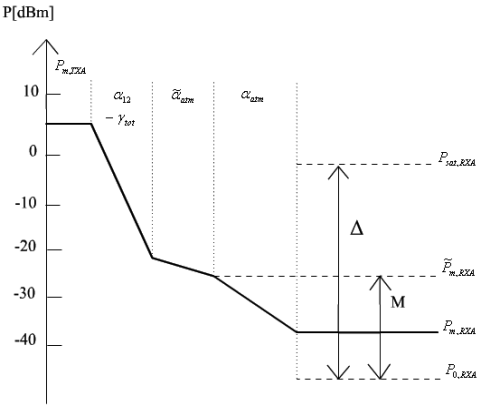


Figure 1. Power diagram for the FSO line.

Identify applicable sponsor/s here. If no sponsors, delete this text box. (sponsors)

B. Attenuation of clean atmosphere

Attenuation of clean atmosphere consists of so-called attenuation caused by the particles in the atmosphere α_{part} and the attenuation caused by turbulence of the atmosphere $\alpha_{turb.}$ The decibel values can be expressed as [5]:

$$\alpha_{part} + \alpha_{turb} = \tilde{\alpha}_{atm} \tag{2}$$

C. Safety edfe

Safety edge is a difference between received power in the clean atmosphere and limit of sensitivity of the receiver. Its value can be calculated as [5]:

$$M = P_{m,RXA} - P_{0,RXA} = P_{m,TXA} - P_{0,RXA} + \gamma_{add} - \alpha_{geom}(L_{12}) - \alpha_{atm}(L_{12}) \tag{3}$$

where γ_{add} is an additional gain, α_{geom} is the geometric attenuation and L_{12} is a distance between the transmitter and receiver.

The resulting value M tells us about margin of a communication link at a certain distance between the transmitter and receiver, it shows what attenuation caused by the weather conditions a communication link can withstand.

Attenuation during bad atmospheric conditions

Safety edge and normalized safety edge took into consideration fixed parameters of FSO communication link and attenuation of a clear and quiet atmosphere. From their results we can determine the maximum attainable distance of link for ideal conditions and the maximum power reserve from which we can still grab losses caused by degraded weather conditions such as fog, snow or rain [10,11].

Fog is the greatest enemy for optical link. It happens because the greatest attenuation of optical link in the atmosphere is caused by in homogeneities whose size is approximately equal to the wavelength of the transmitted laser beam – Mie scattering, which in our case represent the fog particles (0.5 – 2 μm) [10].

III. SOFTWARE PACKAGE FOR THE STATIC SIMULATION OF FSO LINK

According to the input technical parameters of devices, distance and conditions of transmission channel, this program calculates availability or unavailability of communication links in a given environment. The program was created in a development environment Microsoft Visual C# 2008 Express Editions [5].

After opening the software package FSO SystSim an initial window can be seen.

The whole program consists of two basic parts - Static (steady) and Statistics (statistical) model. Switching between these models is done by switching the tabs in Windows applications (*Fig. 2*). The figures show the static model.

Window of static model consists of several parts. On the left side there are options for filling the input data. The right side displays the calculated results. Following figure gives graphical representation of the entered and calculated parameters.

In the program section *Device Properties* the parameters of receiver and transmitter of system are entered manually.

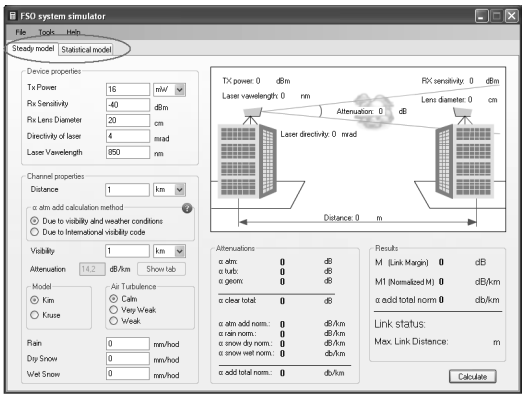


Figure 2. Cards of the "Steady model".

Different parameters are discussed here:

T_X Power - is the mean of laser power, which generates the transmitter. It is the only parameter that can be entered in two different units mW and dBm. This parameter is limited to values from 1.3 to 1000 mW and from 1 to 30 dBm because of the feasibility of calculations.

R_X Sensitivity – in this input field user enters a value of sensitivity of receiver equipment. As in the previous case, parameter is also restricted to values from -10 to 70 dBm. It is necessary to enter a minus sign.

R_X Lens Diameter - This input field contains the value of the diameter of the receiver lens. This parameter is limited to values from 1 to 100 cm.

Directivity of laser – in this input field information about the laser beam directivity are entered. This value is limited to values from 1 until 60 mrad.

Laser Wavelength – it is parameter of system, which tells what wavelength is used. The values are limited to values from 500 to 1600 nm [3].

As noted, each parameter has a limit for entering values. If you enter a value that exceeds the allowed range, the error message is displayed. Change the given value and enter it in the desired range.

Description of database systems

To the original program a new field was added called *Selected system* (*Fig. 3*). This field contains a button that displays the name of the selected system. If there is any selected system, the field contains an inscription *System not selected*. When clicked (in this case *System not selected*) the database of producers can be seen (*Fig. 4*).

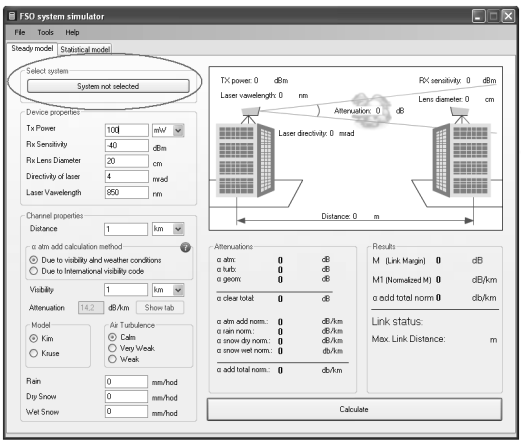


Figure 3. Label of the button to see the Database.

From *Fig. 4* we can see that database consists of two parts. On the left side there is a button to deselect the option and a summary of producers. Right side contains the database of the system of producers. If no producer is selected, field *System of companies selected* displays *Company not selected*. After selecting a certain producer it will show an overview of products, as is shown in *Fig.4*.

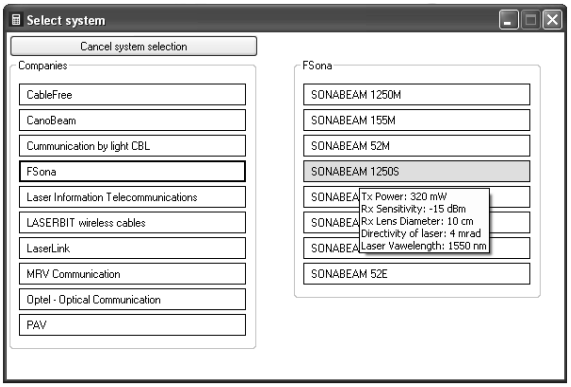


Figure 4. Database of the Producers.

When the system is chosen, parameters are copied into above-mentioned field *Device properties*.

To make the selection of systems easier, there is a possibility to display various parameters in the form of help. It allows us to choose a system based on the displayed parameters. Help appears when the cursor let stands for a moment in any scheme (*Fig. 4)*. Help contains the information above the system that we need for computing (*Fig. 5*).

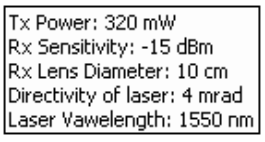


Figure 5. Help for system.

After selecting and subsequently clicking on a system, window for selecting of system will disappear and parameters of selected system can be seen in the *Device properties* and also name of system in the field *Select system* (*Fig. 6*).

After setting the properties of the channel you press a button *Calculate*. On the basis of the parameters of the system and properties of the transmission channel, the program will calculate access of connection. Database of FSO systems in this program is suitably designed to allow easy addition of other producers and their systems *(Fig. 7)*.

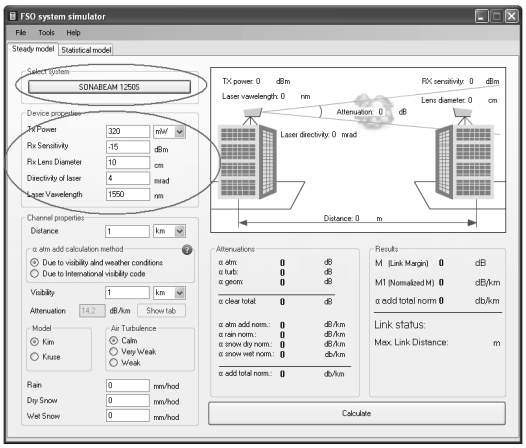


Figure 6. View parameters of the systems with title system.

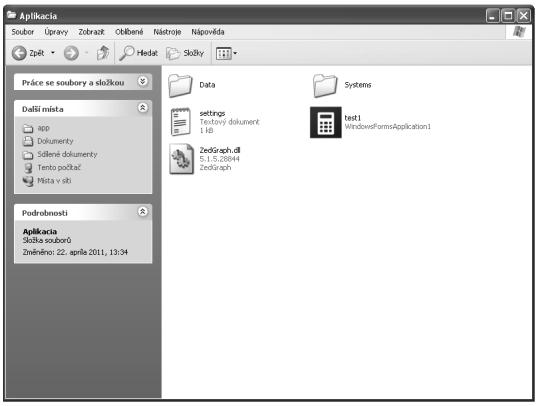


Figure 7. Necessary files FSO simulator.

Entire database is located in the folder *Systems*. If we enter into that folder, all producers for this Simulator display there (*Fig. 8*).

In this new folders for producers can be created and they will be included in the FSO database simulator and therefore their products can be tested. Systems are added to individual folders of producers in the form of .txt files with the given parameters. So, if we create a new producer by creating a new folder with the producer´s name and we want to specify the particular systems we will come into this folder and create .txt files as shown on *Fig 8*.

On *Fig. 9* we see the eight .txt files, 8 specified systems of Fsona company. Parameters and names of individual systems are added to these .txt files as can be seen from *Fig. 10* in the following order: *name of system*, *transmitted power* and its *unit*. Units for transmitted power can be selected and the parameter with unit is displayed in help window, *receiver sensitivity*, *receiver lens diameter*, *directionality of laser* and *laser wavelength*.

Database is ready for further input, either of new producers or existing systems or change of parameters, etc.

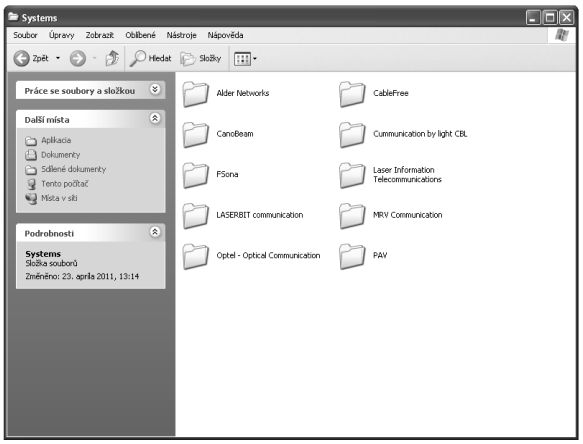


Figure 8. Entering manufactures.

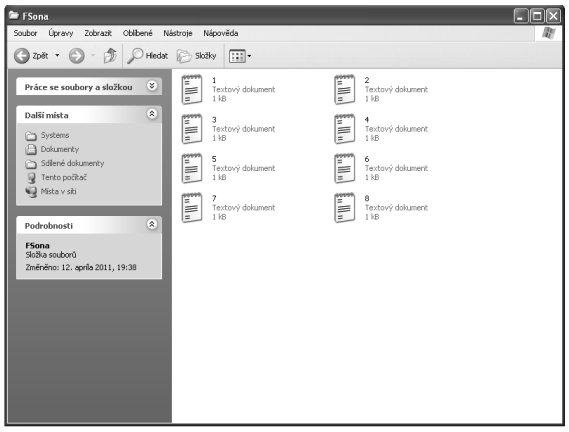


Figure 9. Entering systems of the manufacturers.

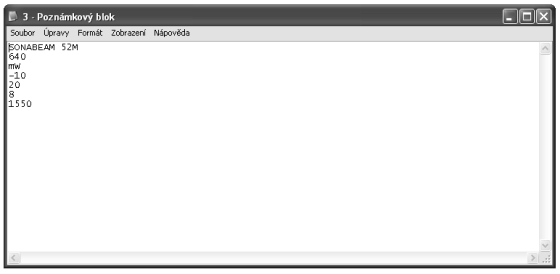


Figure 10. File for input parameters.

IV. EXPERIMANTATION WITH THE STEADY STATE MODEL

As we mentioned above by the insertion of database of specific products of FSO, different systems can be compared with each other under various weather conditions or in order to verify information provided by producers.

In the static model of simulator we set weather conditions in part *Channel properties*. *Fig. 11* shows possibility to set conditions and to add the item *α atm add calculation method*. If you chose the option *Due to visibility and weather condition* we can simulate different losses, caused by rain or snow. We can also simulate turbulence by the selection of Kim or Kruse model (*Fig. 11)*.

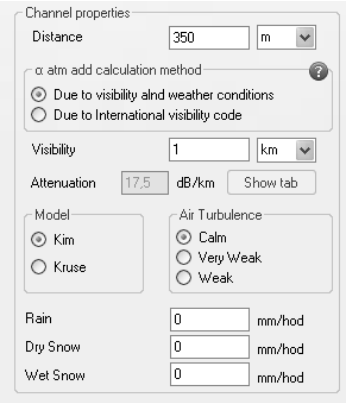


Figure 11. Entering the atmospheric conditions and visibility due to weather condition.

If you chose the option *Due to International visibility code* (*Fig. 12*) you can enter required attenuation to the field *Attenuation* or you can select *Show Table* which will subsequently display a table (*Fig. 13*) from which the loss can be determined. Other fields except turbulence remain hidden. This means that the final attenuation is entered directly to the field attenuation.

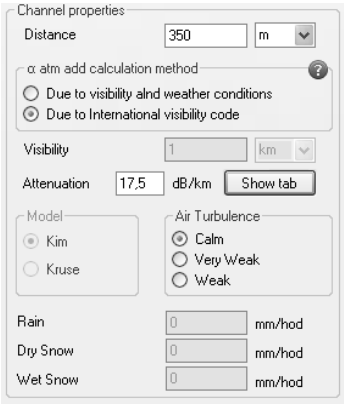


Figure 12. Entering the atmospheric conditions for due to international visibility code.

Visibility And Attenuation In Various Weather Conditions

Weather conditions	Precipitation		mm/hod	Visibility	Attenuation (dB/km) at λ = 785nm
Dense fog				0 m	350
Thick fog				50 m	339,6
Moderate fog				200 m	84,9
Light fog				500 m	34,0
Very light fog	Snow	Storm	100	770 m	20,0
				1 km	14,2
Light mist		Strong rain	25	1,9 km	7,1
				2 km	6,7
Very light mist		Average rain	12,5	2,8 km	4,6
				4 km	3,0
Clear air		Light rain	2,5	5,9 km	1,8
				10 km	1,1
Very clear air		Drizzle	0,25	18,1 km	0,6
				20 km	0,53
				23 km	0,46
				50 km	0,21

Figure 13. Table attenuation of weather conditions.

Comparison of calculation with the reported parameters

To test the reported parameters we have chosen *Due to International visibility code* method. In this method we can exactly specify the required attenuation that is intended by producer. For a test we have chosen MRV company and products TereScope 5000 and TereScope 10GE. The producer indicates the availability of links for different attenuation in *Tab.I*.

TABLE I.
TABLE OF DISTANCE PROVIDED BY THE MANUFACTURERS

Attenuation	Terescope 5000 Distance (m)	Terescope 10GE Distance (m)
3 dB/km	5500m	1000m
10 dB/km	2700m	600m
30 dB/km	1200m	300m

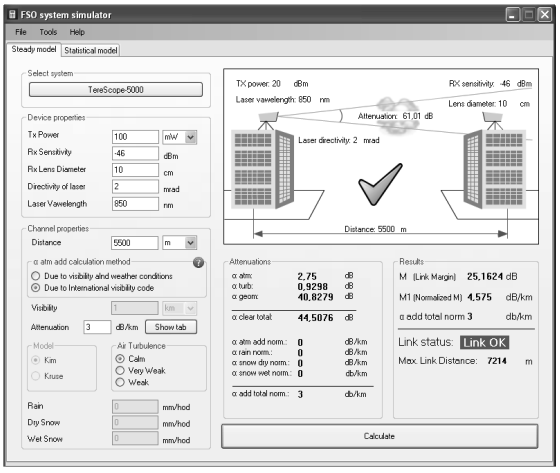


Figure 14. The output of the Steady model.

After the selection of the system Terescope 5000 and setting the attenuation 3 dB/km, what represents clear atmosphere, Simulator calculated the maximum communication link availability to 7214 m for low turbulence (*Fig 14*). With increased turbulence, availability of FSO line is falling. With turbulence *Very Weak* availability falls to 6380 m and 2600 m for the *Weak*.

TABLE II.
TABLE OF RESULTS FROM THE FSO SIMULATOR

Attenuation	Terescope 5000 Distance (m)	Terescope 10GE Distance (m)
3 dB/km	7214m	2313m
10 dB/km	3154m	1274m
30 dB/km	1339m	637m

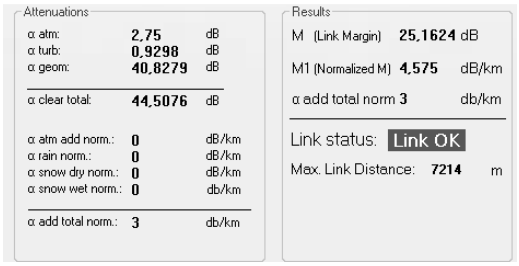


Figure 15. The output from Simulator.

On *Fig. 15* we see simulator´s output. In the part *Attenuations* we can see the calculated attenuation and in the part *Results* we can see total attenuation, safety edge and a decision about availability of FSO line. Other calculations for Terescope 5000 system and Terescope 10GE system are shown on *Tab. II*.

By the comparison of *Tab.I* and *Tab. II* we can see that the calculated values differ mainly in TereScope 10GE where distances are twice as big in given attenuation as indicated by the producer. In TereScope 5000 system deviations are not so great, for attenuation 10 and 30 dB/km the deviation is 300 m, for pure atmosphere it is probably around 1700 m. By calculations we found that the TereScope 10GE system responds better to increase of turbulence than TereScope 5000 system because the increase of turbulence availability caused the decline by only a few hundred meters. On *Very Weak,* access communication links declined by only 100 m and on the *Weak* by 300 m.

Comparison of different systems among themselves

Comparison of systems allows us to understand how at selected atmospheric conditions the system will behave. According to the calculations we can determine which product is suitable for the environment and which is not.

For comparison we choose systems CableFree Access A1000, Sonabeam 52E from Fsona PAV Light and Gigabit. We set the same atmospheric conditions for all products, Kim´s model for turbulence *Calm* and visibility 1 km for distance between the transmitter and receiver 1 km. Results are shown on *Fig. 16, Fig.17* and *Fig 18*.

It follows that the first two systems are suitable for the environment while the system PAV LightGigabit still has some drawbacks and system Sonabeam 52E is located on the edge of functionally. CableFree Access A1000 system is not suitable for the environment.

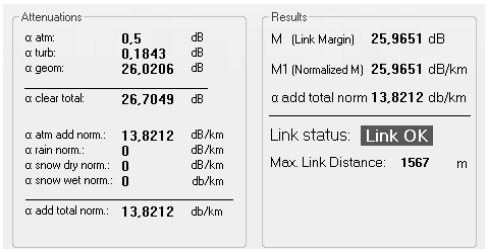


Figure 16. Output steady model for PAV Light Gigabit.

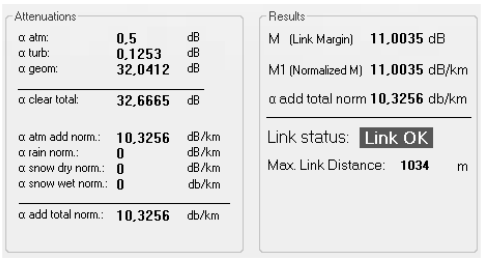


Figure 17. Output steady model for Sonabeam 52E.

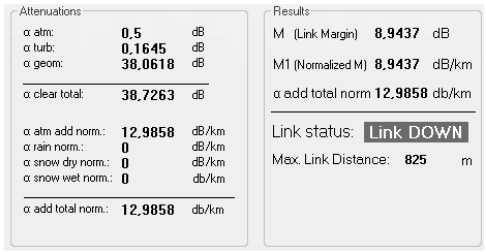


Figure 18. Output steady model for CableFree Access A1000.

V. CONCLUSION

Optical communications by free environment are an evolving technology which rapidly spread from objects in need of reliable backup and high speed transmission (banks and large management companies) through various institutions located in the densely populated and built-up areas to the internet service provider where the market is pushing the need for increasing the permeability data lines. For quality and efficient design of FSO systems is necessary to know the details of each element and see the facilities connection between changes in equipment parameters, change the properties of the transmission channel and the resultant effect. Simulation of optical communication free environment is an essential tool in designing and experimenting with such devices.

The user determines of the results the values of each attenuation and final outcome including theoretical design of maximum attainable distance. We can expect theoretical behavior of the line during different atmospheric conditions occurring during all seasons (fog, rain, snow) by simply changing parameters in the program.

ACKNOWLEDGMENT

We support research activities in Slovakia / Project is co-financed from EU funds. This paper was developed within the Project "Centrum excelentnosti integrovaného výskumu a využitia progresívnych materiálov a technológií v oblasti automobilovej elektroniky", ITMS 26220120055 (50%). This work was partially supported from the grant VEGA 01/0045/10 (50%).

REFERENCES

[1] Kim I., McArthur B., Korevar E.. Comparison of Laser Beam Propagation at 785 nm and 1550 nm in Fog and Haze for Optical Wireless Communications. In Proceedings of SPIE. Volume 4214, Boston, USA, 2001, pp. 26-37

[2] Chabane M., Alnaboulsi M., Sizun H., Bouchet O.. A New Quality of Sevice FSO Software. In Proceedings of SPIE. Strasbourgh, 2004.

[3] Kolka Z., Wilfert O., Kvicela R., Fider O.. Complex model of Terrestrial FSO Links. In Proceedings of SPIE. Volume 6709, 2007.

[4] Bouchet O., O´Brien D., El Tabach M., M. et al.. State of the Art-Optical Wireless. Deliverable D4.1, ICT – Omega, 2008.

[5] Hranilovic S.. Wireless Optical Communication Systems. Spring, USA, 2005.

[6] Kvicela R., Kvicera V., Grabner M., Fiser O.. BER and Availability Measured on FSO Link. Radioengineering, Volume 16, No. 3, 2007.

[7] Muhammad S. S.. Investigations in Modulation and Coding for Terrestrial Free Space Optical Links. Dissertation work, Graz, March 2005.

[8] Bouchet O., et al.. Free-Space Optics, Propagation and Communication. In Proceedings of ISTE, 2006.

[9] Majumdar K. A., Ricklin C. J.. Free-Space Laser Communications. Spring, Berlin, 2007.

[10] Bloom S., Korevar E., Schuster J., Willebrand H.. Understanding the Performance of Free-Space Optics. Journal of Opt. Netw.. Volume 2. No. 6, 2003, pp. 178-200.

[11] Alnaboulsi M., Sizun H., deFomel F.. Fog Attenuation for Optical and Infrared Waves. Journal of Optical Eng.. Volume 43, 2004, pp. 319-329.

Analyse of selected properties of websites in the cities

Igor Bandurič
Department of Applied Informatics
Faculty of Economic Informatics
University of Economics
Bratislava, Slovakia
Email: igor.banduric@euba.sk

***Abstract*—Nowadays the local municipalities have to deal with the necessity to have own web page. This necessity rise from fact that there is increasing pressure to have better public control over self government institution. This is most influenced by legislative act about free access to information num. 211/2000 Z.z. [1] Because local municipalities are constrained by budget and human resources there can be a tendency to find solution which is most effective. This paper deal with analyse of selected web pages attributes of the cities in Slovakia. The attributes are mostly from technical background of web pages. By analysing this attributes we want to find out if cities are using common technologies or common vendor.**

I. Introduction

There are 138 cities[1] in Slovakia [2]. All cities have own webpage, which was not true several years ago. According to [3] in 2003 there were 26 cities without webpage. However some of the cities we analysed have been titled as *unofficial* which probably means that web page is maintained by some public organisation and not by city itself.

The creation of such page is in case of city influenced by several factors, which are usually not present in private sector. These factor can influence technology used for page creation and in future can be a barrier for required changes or on the other side can be very helpful to adapt this changes.

Factors influencing webpage creation

1) Legislative acts
2) Cost of software licenses

A. Legislative acts

Nowadays there are several legislative acts which prescribes how the webpage should look like and what it should contain. The most important is legislative act number MF/013261/2008-132 [4] about standard for Information system in public administration. This act prescribes standards, which must be fulfilled. Aim of these standards is mostly to obtain accessible web page for handicapped people. Failure to fulfil these standards can be persecuted by penalty. These standards are periodically monitored and reported. These standards can be viewed as quite strict.

As mentioned in [5] these initiatives could be quite contra productive. Small city doesn't have capacity to change the existing web page so fast. And this can lead to situation that city or village decided not to have webpage at all rather than have simple page which can be persecuted by penalty (as we mentioned, this is not the case for Slovakian cities - all of them have a webpage).

There is also number of unofficial monitoring which are published. Together with official monitoring they develop a pressure on local administration.

From the beginning of year 2011 local municipalities have to disclose their contract and invoices. These should be disclosed mainly on their webpage.

These official requirements are common for all cities. Accessibility can be solved by good *content management system* (CMS), which will follow all required standards. The same way can be used when disclosing contracts and invoices. Good technical solution can help city disclose required documents on the webpage.

B. Cost of software licence

The creator of webpage have basically two possibilities when creating cities' page. One is to choose commercial products and the other is to choose open source. The same chooses are when deciding what kind of server to use. Of course we cannot say from the analyse if this decision was made by web provider or by city administrative by itself. Using open source doesn't automatically mean that creating web page is for free, but at least there are no additional payment for using commercial product.

II. Analysed attributes

The aim of our analyse is to find out if this two mentioned *pressures factors* have some implication on technologies used in the background of the cities webpages.

We analysed these attributes:

1) Language used
2) Server which serves requests
3) Generator
4) Author

For the analyse we created crawler script written in *Groovy* language. Script firstly crawl wiki site [6], which contains list of cities and grab the web address. In the second step has script

[1]Including also cities which doesn't have own district and are part of other town district

visited every grabbed url and analyses the response. Analysed was only the home page not the rest of the page.

We did also compare our result with online service for Slovak domains analyse [7]. The script for this online service is written in *Ruby* language and provides a slightly different result while it counts on fact that some generators implied the used language. This script also doesn't look for the author of the page. Analysing the author of webpage, as it turns out, has interesting result.

A. Language used

When crating webpage one will probably mostly choose PHP as language for the dynamic parts. This is true also because most of CMS have been written in PHP. This expectation was confirmed by analyse shown in Figure 1. We used for discovering language used the http response header `X-Powered-By`. We didn't make any assumption that some other attributes can imply language (i.e. used CMS can imply that). Comparing to result provided by [7] only PHP language have better result. There were no pages using *Ruby* or *Java* language. However we would expect to show more *Java* powered sites as soon as the web page will become more application than simple webpage.

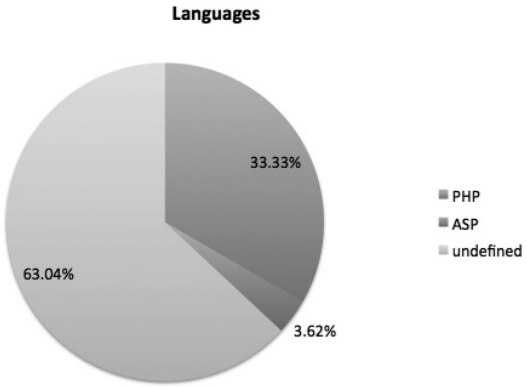


Figure 1. Language usage

B. Server which serves requests

We used for discovering language the http response header `Server`. Server detection was successful in more cases than Language detection. Some of the server also provided not only the name of server but also the version and installed plugins. The mostly used server is Apache Figure 2. Sometimes the Apache server is only acting as load balancer and is only as Façade for server crating pages. This case will need more analyse but wi do note expect that it change results significantly. Also Apache is mostly used, we can see that increasing popularity of Ngnix server has reached also this market. Both Apache and Ngnix server are open source and there is no additional cost for using them. On the other side is the Microsoft IIS server which is distributed under commercial licence. The popularity of IIS according to our analyse is the same as Ngnix server.

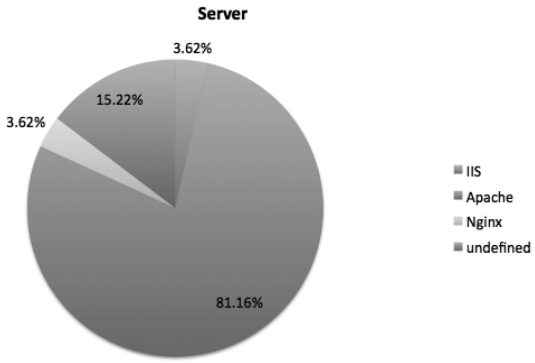


Figure 2. Servers

C. Generator

By generator we understand software which generated page. This kind of software is mostly some kind of CMS. Detection of generator was done by analysing `<meta>` attribute in the generated page. `<meta>` attribute named *generator* should contain name of such software. We also used some other mechanism to detect generator such as presence of known generator name in the content of home page. This name could be included in comments, path for cascade styles or path for external script. Most pages didn't contain this information so in Figure 3 we only show these which does. The pages, which doesn't contain *generator* information were found in more then 80% cases.

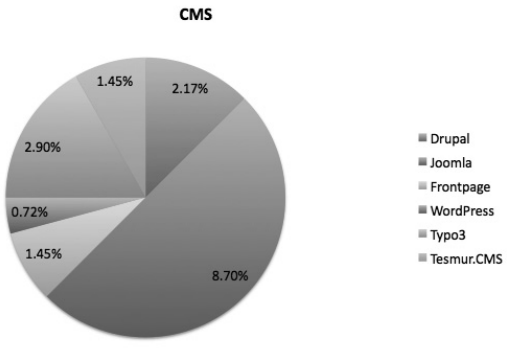


Figure 3. Generators

Most of these CMS are new. So they should have no trouble to follow current webpage standards (including those mentioned in subsection I-A) The only exception of these is Frontpage from which one was only in version 3.0 which was distributed in year 1997.

D. Author

Detection of author was done by analysing `<meta>` attribute in the generated page. `<meta>` attribute named *author* should contain name of webpage author. More than 50% pages contained this information. More than 25% of pages have concrete author, which was mostly unique (only two authors have been detected on more than one page). These authors are noted as *named* in Figure 4. Those which doesn't have this meta tag are named as *unknown*. The most pages, where the author is known, has one common author - *webygroup*. One company develops more than 25% of cities' webpages.

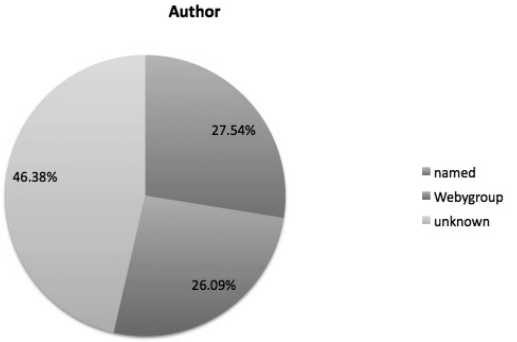


Figure 4. Authors

While webpages, which are developed by *webygroup* doesn't detect software used (CMS) we can alter the result for Generators in subsection II-C which will change the result and mostly used CMS will be one developed by webygroup with more than 25% share followed by Joomla with slightly more than 8%.

III. Discussion

The webpages which does't show server name, language used or CMS used can be divided into two groups. The first group is one in which one of these attributes is simply not shown while there is no web language or no CMS used. The second group is group of pages where this attributes were intentionally turned off (mostly by security reason)

The expected tendency to use one common solution (webygroup on the first place and Joomla on second) which will serve slovak requirements has several implications. We will discuss several of them.

The first is that there is still some place on the market to obtain own market share. There can rise up new companies which will satisfy the demand. On the other side there should be some careful observation if companies with big market share doesn't have some unallowed competition advantages.

The second is opportunity to develop custom plugins to existing open source CMS rather than develop new one. Because of open source characters mentioned CMS this can be good platform for students projects.

IV. Further analyses

We would like to continue in our research and analyse more technical attributes of webpages. We expected that mentioned pressures will lead to more standardised solution from which *webygroup* is good example. It will be also interesting to see the registrar and the hosting of the webpage which can be detected from *sknic* databases.

In the further analyse what we would like to do is to crawl with developed script webpages of all local administrations - that means including villages which were not included in current analyse. Including villages can be challenge while some of them doesn't have a webpage or webpage is not actual. Beginning in the past, there were some project which tried more or less successfully to offer space on web servers for small cities and villages. One of the mostly known in the past was project ISOMI [8].

V. Conclusion

Todays monitoring analyses mostly accessibility attributes for handicapped people and data formats used on pages. We did research on slightly different attributes which are more technologically oriented. As mentioned we analysed home pages. We downloaded only the first page of the cities and we didn't visit other pages. Analysing more pages will gave slightly better result while some cities (i.e. Bardejov) still use the, so called, *Welcome page*. Because of detecting algorithm used in crawler script the result are not one hundred percent correct but they still provide good picture of the current situation.

One can expect that because most of pages use on technological solution they will have no troubles to follow standards and legislative act. However the opposite is true. As mentioned on the home page of webygroup company [9] the last result of monitoring has shown that cities which used webygroup CMS has been among first 3 places but also among last 100. Which means, as webygroup notes, that not only technology is needed but also people which will use this technology correctly.

We can also expect that due to progress in digitalisation there will be more pressure to good technological solution. Demand for electronic forms is one of them. And as mentioned in previous paragraph the technology is only one of the pair, the second si human.

References

[1] "Zákon č 211/2000 z.z. o slobodnom prístupe k informáciam," 2000.
[2] "Zoznam miest v slovenskej republike podľa krajov a okresov," Feb. 2004.
[3] M. Bačík, "Informatizácia a internetizácia vo verejnej správe," *Geografické aspekty středoeuropského prostoru, Geografie XIV, MU Brno, Pedagogická fakulta, str*, pp. 66–70, 2003.
[4] "Výnos ministerstva financií slovenskej republiky z 8. septembra 2008 č. MF/013261/2008-132 o štandardoch pre informačné systémy verejnej správy," Sep. 2008.
[5] V. Regec, "Prístupnosť webových stránok v slovenskej republike," Mar. 2009.
[6] "Zoznam miest na slovensku - wikipédia," http://sk.wikipedia.org/wiki/Zoznam_miest_na_Slovensku, Sep. 2011. [Online]. Available: http://sk.wikipedia.org/wiki/Zoznam_miest_na_Slovensku
[7] "Štatistiky domén," http://www.statistiky-domen.sk/, Sep. 2011. [Online]. Available: http://www.statistiky-domen.sk/

[8] "ISOMI, a.s." http://www.isomi.sk/, 2007. [Online]. Available: http://www.isomi.sk/
[9] "Webygroup - komplexné služby pre váš internet," http://www.webygroup.sk/, Sep. 2011. [Online]. Available: http://www.webygroup.sk/

Cloud Computing Business

Matej Kultan
School of Science and Technology
Aalto University
Espoo, Finland
E-mail: matej.kultan@gmail.com

Abstract— **The Cloud Computing Business has a lot of points of view. The press, investors, marketing agencies and other organizations review this topic to find benefits, risks from economical, technological and financial overview. To provide an overview and analyze Industrialized IT environment in this document, there were used methods like Business Canvas Model and general marketing research.**

Keywords- Industrializiton of IT, Business Canvas Model, Key resources, marketing

I. INTRODUCTION

We have been looking for an appropriate model to describe what are the main topics related to The Business in Cloud computing. To approach the best description of the topic, we have decided to use Business Model Canvas. The Canvas model is suitable for a general overview of needs and relationships between Value Propositions, Costumer Relationships, Channels of distribution to the Costumer having regard to observe Key Activities. These activities imply need of carefully chosen Key Resources and Partners to achieve a standard and quality product brought to the end costumer. The final product needs appropriate financial analysis and thus there is also need to analyze Cost structure and Revenue streams to let the Business work properly.

All mentioned topics are included into the Poster and they were detailed discussed during the presentation session on 1st of December, in the main hall of Computer Science building. The main point of the presentation was to expose main key values and problems that are related to Industrialization of IT.

II. THE VALUE

The value behind cloud computing is the need to industrialize the storage of data to be able to manage it as it becomes bigger it year. The desire to be interconnected of the users and future uses like making the process work some computer may not be able.

There are businesses that achieve value creating a fictitious need in the user. We can see example of this in wide consumed goods like cologne. Almost all the advertisement of colognes tries to convince the costumer, they need the cologne to become attractive to the other sex. They attach desired attributes like appealing body/beauty to the product using models to perform the spots. This is in the best scenario a exaggeration and in the worst a completely lie.

There are others that present a more sincere or direct value/need. This is the case of, for example, clean products, whose makes arguments to sell are "It is easier to clean with this".

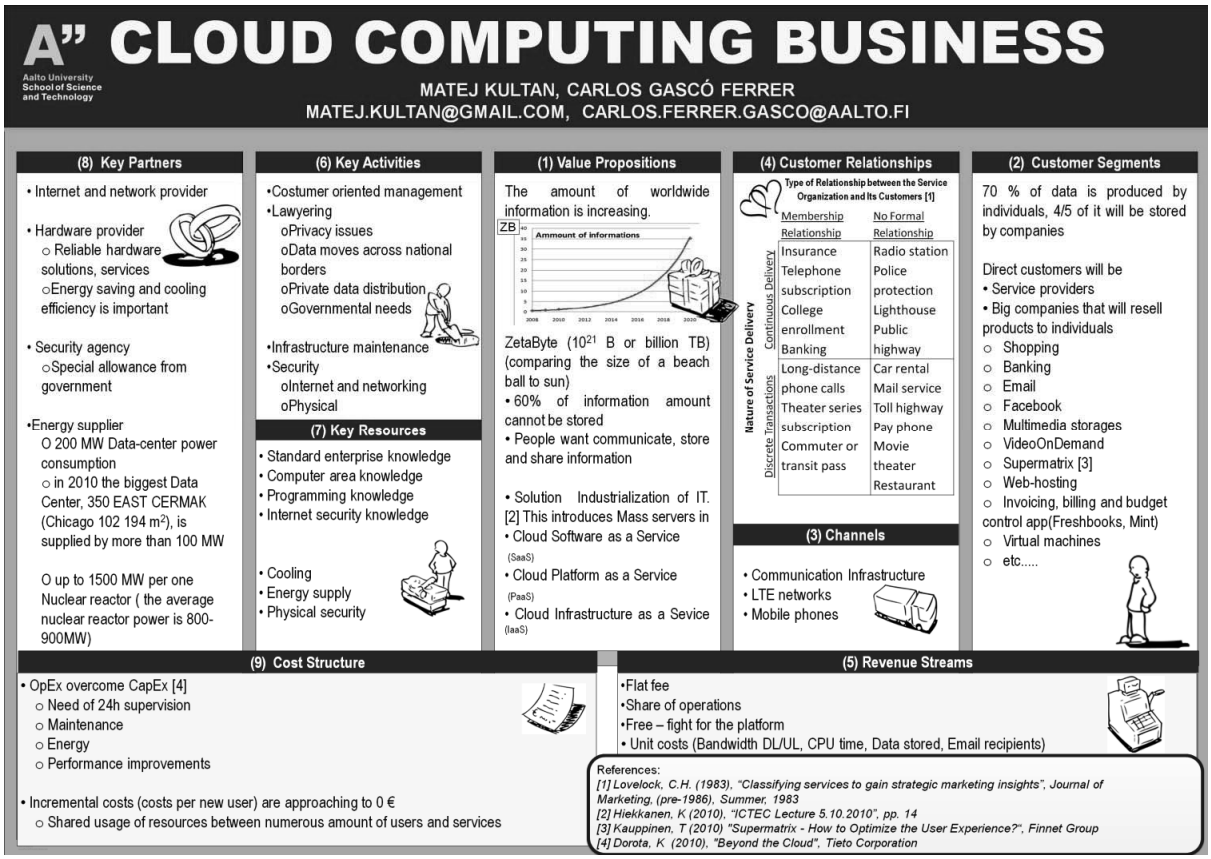
There is a third group that work with an even more direct value. I will call it value through real need. Here we have business that allow the user to perform or solve problems will be impossible to overcome without the product. This direct value feeling is typical of business with little competence so on new business field. As a real need is fulfill by more enterprises the value they sell transit form the original need to fictitious previously name in an attempt to differentiate the own good from the rival companies'. For example, it is quite obvious a car has a very strong direct value, it allow you to travel, so if cars were new products in the market talking people to buy one would be as easy as telling them what a car is for. However , now that a car is a good almost taked for granted, and there are many companies that fulfill that need we can see them making spots aiming to other values/needs to differentiate them from others.

We, as cloud service providers, are more in the third kind. Depending of the level we are in the cloud service ladder, IaaS, PaaS or SaaS, it will vary. As closer we get to IaaS low level, that rent raw machines to big companies we will get more close to direct/true need. And as we get upper in the layer to SaaS we will require more, marketing tricks to convince users to choose our solution.

III. THE TARGET CUSTOMERS & USERS

The target user will be almost every computer user, from professionals to casual. Professional may know about cloud computing possibilities and so on an explanation of why they are suitable target is not required. For the casual users we will have to realize a lot of service they are already using are cloud computing ones, like email, web navigation, social net. They are sold to them as simple final services without the technical stuff about cloud computing, although they are it.

To provide this final services a company will need more technical cloud computing services like servers for example that are bought from other company. Here we can see the ladder of cloud computing services we have previously mention. Someone rent raw machines, or capacities, this is the



Picture 1, Business Canvas Model **[1],[2],[3],[4]**

IaaS. Someone buy them to install configurations in them and sell them as more specific virtual machine, ubuntu machines for example, this is the PaaS. And some one rent the virtual machines to provide develop software, SaaS. Not all of this steps are need and a company can take over more than one.

So, apart from IaaS, the rest are providers of consumers as well, everything orchestrated to deliver the final service to the final user, everyone computer user.

IV. THE CHANNELS

Since this is IT and there is no physical good to be transport to the user, the channels will be the IT communication infrastructure that allow the traffic of data and provide the service. As the channel becomes more sophisticated, more features are allowed and so on the service improve.

V. THE CUSTOMER RELATION

The level of relation, fidelity or interaction between the providers and the consumer while vary depending of the value to reflect it. So on, IaaS providers, focus on selling capacities to solve real need while surely have a more sterilize relation with the consumer since they do not need to appeal to the feeling to sell. In the other extreme SaaS with more rival companies will have to appeal to fictitious needs of the users, and try to cultivate the loyalty them, this will be reflected as membership relation that aim to maximize the human dimension of the deal.

VI. KEY NEEDS AND RESOURCES

When venturing in an business enterprise you will always need a common activities, that we can refer as business stuff. It is not the point of this essay to treat about them so let's just say that you need them.

Apart from them you will have specific need of your field, in our case every step of the ladder will need the services of the bottom one.

If you are a SaaS you will need a PaaS provider and business stuff oriented to your business. You while probably need the more advance marketing division of the all levels since you are in a more competitive market. You are aiming to simple user computers so you cannot win their hearts with technical superiority but with the "general impression" of being best, cooler, fancier…etc. Also individual treat of the

users is not possible, nor are individual contract. This is like many other mass businesses.

On the other hand and being much more interesting you have the IaaS. The bottom layer that supports everything. They are one of few that can serve the resources and capacities needed by other steps of the ladder. They treat with big companies, make specific contracts and deals that go though tough negotiations. They are like the paradigm of business to business system. We will focus on their special needs.

A. The electrical need:

This huge data-center site needs a lot of energy. One of the best nuclear plant we have produce 1500 MW. These IT bastions consume around 200 MW. So a nuclear plant will only be able to support seven of these centers. You will need to be near a power plant.Yes, you are the bottom and king of the cloud computing ladder but you are only a second step in the whole society. You rely on energy.

B. The cooling:

The heat produced by these servers is so that you need huge amounts of water to cool it. You need to be near a river. This is not so difficult since the nuclear plants you are pursuing also need water and gather around rivers.

C. The Security:

The safety of the computing and data has such high importance as its reliability. To assure the security of the cloud computing product there is need to create whole system. We can consider this topic scalable into 3 parts:

1) Physical security

With the amount of data you are storing, you have to protect your servers, a lot of companies rely on you and if, for example, a terrorist blow you up there will be a very negative year for a lot of people. The value of these centers are so that the security perimeter radios commonly have kilometers. You are going to need special allowance of the government to gather the military forces.

2) Corporate Security

To assure process safety, it is necessary to create also corporate environment that includes mechanisms compared to errorless program. It is necessary to spread the power of personnel by strict logical policy, and spread the territorial and partnership dependence as well. This is really hard task to achieve, because every security process decreases the flexibility of the provider.

3) Network and IT infrastructure Security

This topic includes all imaginable processes like security cryptography, data integrity, confidentiality that are about to be assured by Cloud Computing supplier as its standard service. Easily explained, nobody will leave confidential data on a non-secure environment such a platforms with backdoors etc.

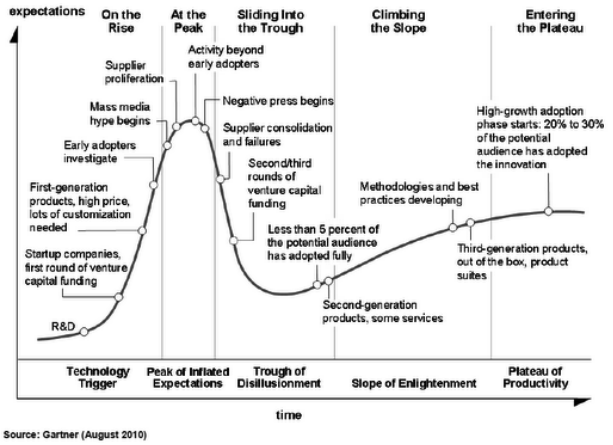
D. The lawyering:

The information trespass borders from and to your servers you will have to have special care with the difference laws of the countries you serve. What it is illegal to be stored there maybe is not here.

We can see it is not easy to be IaaS and a lot of capital, resources and contact are needed.

VII. THE MARKET RESEARCH

It is very important to monitor market needs. Following the slope, at this moment in the end of 2010 we can consider the position in time somewhere in the Peak of Inflated Expectations as there was already a lot talk about Cloud Computing and some activities and early adaptations. There are already tries of standardized ways to access Industrialized IT although there is still a huge space for global and local users.

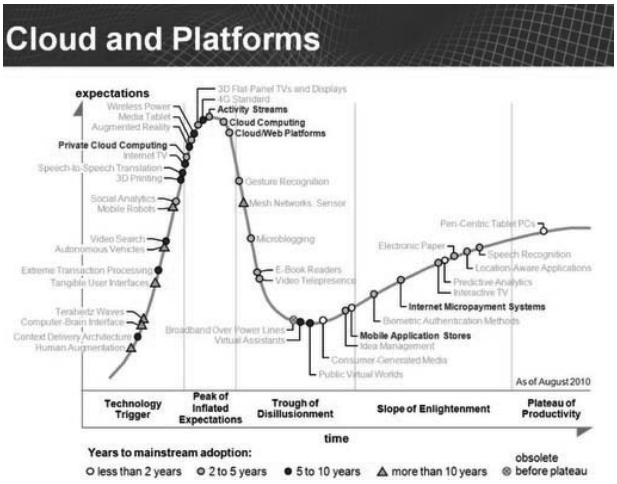


Picture 2, Product time-expectation slope [6]

The market will realize the real power of Cloud Computing in next few years with a more realistic view in timeline and putting real goals to be accomplished. What is more important are the technology expectations comparing to other challenges. The Sci-Fi projects like Human Augmentation, Mobile Robots have a low expectation as they are long-term project not-available in market following 10 years. More importance is given into 3D technology, Mobility Services as they are more probable to deploy in 5 to 10 years. The issue with Cloud Computing is nowadays a hot topic because the

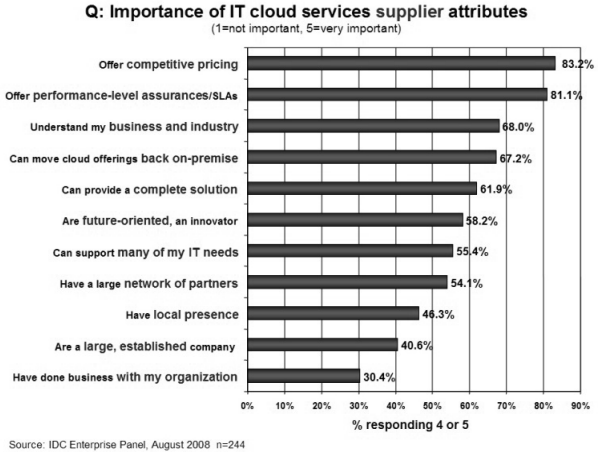
technology level necessary for deployment is available and practically ready to be built in short-time.

Problem still persists in the service reliability and security, as from the May 15, 2009 was exposed, the failure of the Google Cloud had irreversible effects on loosing users personal data and that is still the reason why the investments and customers will start to look more from the top on this technology. The reliability is an end-less issue and cannot be assured in any business cases, only it can be guaranteed by a statistical tolerated mean value that will be decreasing by time as the Cloud product matures.



Picture 3, Cloud and Platforms expectation slope [6]

According to market analysis of IDC we can still say that customers have still need to follow the competitive pricing, performance-level assurance and they are expected to understand business and industry needs. Complete solution is also demanded and that implies that the existing customer already knows "what and how" to use cloud computing to its needs. Subjective needs as supplier size, partnerships and local presence have not a such importance.



Picture 4, Cloud computing customer survey [7]

The average customer behavior is following the expectations of the whole idea of Cloud computing, that is "out-sourcing" of resources that are out of the core business and generalize the whole supplied IT department by a management and economical value.

VIII. Conclusions

From the point of Economics, the Industrialized IT needs knowledge IT on different levels from networking up to system and service development, process automation and company management, security, power engineering etc. Not only continuous need of energies, but also lawyering support, and security are exposing the need of high-qualified human and financial resources. So the problem with the Business in the Cloud Computing is not about to build complete static solution. The problem is based on well combination of static and dynamically elements to maintain the company developing 25 hours/8 days per week providing flexible solutions for masses.

Generalizing whole topic, the operational expenses (Opex) is overcoming high capital expenses (Capex).
When whole company runs in a continuous mode, the revenue depends a lot on marketing by approaching costumer efficient-pricing models. The revenue is calculable by total system capacity (Network, Data storage, Computing power, additional services etc.) and its statistical load thresholds. Any further incremental investments/cost are comparing to Opex low and are smoothly increased by the number of customers and their load needs.

Concluding, the growth and further company investments comparing to demand needs is linear with a low index of growth that allows a relatively cheap service for masses of end-users.

IX. References

[1] Lovelock, C.H. (1983), "Classifying services to gain strategic marketing insights", Journal of Marketing, (pre-1986), Summer, 1983

[2] Hiekkanen, K (2010), "ICTEC Lecture 5.10.2010", pp. 14

[3] Kauppinen, T (2010) "Supermatrix - How to Optimize the User Experience?", Finnet Group

[4] Dorota, K (2010), "Beyond the Cloud", Tieto Corporation

[5] E. Barrera. (2010, December 14) Cloud Computing Diversity Needed. [Online]. Available: http://www.adotas.com/2009/05/cloud-computing-diversity-needed/

[6] K. Troffell. (2010, December 14) Emerging Technology Hype Cycle 2010: What's Hot and What's Not. [Online]. Available: http://bimehq.com/cloud-computing/analysis-gartners-emerging-technology-hype-cycle-2010-hot/

[7] F. Gens. (2010, December 14) Clouds and Beyond: Positioning for the Next 20 Years in Enterprise IT. [Online]. Available: http://www.slideshare.net/innoforum09/gens

Complex Support of Functioning and Development of Higher Education on The Basis of Information Technology

Yahya Buchaev, PhD, Professor, Rector,
Karahan Radjabov , Ph.D., associate professor, Dean of the faculty of "Applied informatics"
Makhachkala, Dagestan State Institute of National Economy under the Government of Republic of Dagestan (DGINH), dginh@yandex.ru

Abstract – The article deals with the problems of implementation of IT into the process of development of the institute.

Keywords- education, development, problems, solutions, information Technology, MIS.

The strategic goal of public policy in education (that is: increasing the availability of quality education which meets the requirements of innovative economic development, the contemporary needs of society and every citizen) is implemented in all federal districts and regions of Russia. The realization of this objective requires addressing such priorities as:

- Provision of innovative character of education.
- Modernization of education institutions as instruments of social development and creation of a modern system of lifelong education, training and retraining of professional staff.
- Formation mechanisms for assessing the quality and demand for education services to employers; participation in international comparative studies.

Progress in these tasks is determined by the new ideas and technologies. Russian universities have to find an acceptable solution to the fundamental problem of sources to cover the cost of student learning (which is especially important to Dagestan), while maintaining equal access to Higher Education. The relevance of curriculum and teaching quality to new needs of an economy, based on knowledge, is an important and urgent task.

It is no accident that new educational standards are being adopted at this stage. Adequate and prompt response of universities to these changes is an important factor for success. All these trends are reflected in the Dagestan schools that effectively use information technology both in management and in the educational process.

The specificity of the Dagestan State Institute of National Economy (DGINH) is that, organizationally and functionally, Institute represents an interconnected educational complex of units, implementing programs of HPE (25 majors), ACT (24 degree) and NGOs (6 specialties). All structural units widely and consistently apply the new information technologies at all stages of preparing specialists (bachelors, masters).

ACS successfully operates in the institute. Activities of such departments as human resources, chancellery, accounting, educational and scientific departments are automated at the level of administration.

Client-server module "Student" ACS "Octopus" is introduced and widely applied in practice at the level of deans of faculties and departments to automate the process of enrollment of students, keep records of students' contingent, also control the learning process and manage the tasks; such as tuition, work with the recruiting office, issue of diplomas, transcripts, personal cards of students and formation of archives.

Subsystems of software package - "Admissions," "Training Accounting", "Express-schedule", "Diploma-standard" and "Electronic Library" are adapted to all organizational units implementing training programs for the ACT and NGO.

The processes of formation of curricula of all educational programs of the Institute, licensing and state accreditation of the university are fully automated with software "Comprehensive assessment of the university." Process is supported at the programmatic level by the use of the software package *GosInsp*.

At this stage of the education system development, intensive research and practical work is carried out in many regions of Russia on creation of

regional training centers with the use of distance education technologies as part of the Concept of creation and development of a single system of distance education in Russia.

Pedagogical staff of DGINH actively works in this direction. Employees have gained a good foundation in the form of electronic teaching materials for the majority of academic disciplines; corporate network is formed, the site of the Institute is developed and filled with information; fresh information and educational technologies are used effectively in the learning process. DGINH has well-trained scientific-pedagogical and engineering personnel and possesses good technical resources. To raise the educational process at a higher level administrative and academic staff of DGINH completes works on developing electronic academies at the Institute under the leadership of Microsoft and Oracle.

File server and distance learning technologies are formed on the basis of a computer network.

Electronic educational resources (both acquired and authoring) are placed on a file server in one of the selected partitions.

Electronic resources for educational purposes - the test tasks, lab workshops, assignments for independent work, also educational courses and tests for monitoring, have been developed. In this way a directory of electronic educational resources was created. Access to the educational information resources is realized from computer classes that are connected to corporate networks or the Internet.

Another effective way of influencing the quality of training is the development of test quests in all academic disciplines in the framework of "Examiner" and "Test Designer."

These developments are an important addition to educational and methodical complexes (EMC) created at the institute and published in accordance with the recommendations of the Scientific Council of the Institute, and placed on the server of the Institute with access to them both for learners from their workstations, and through Internet.

In the process of technical re-equipment lecture halls, computer labs are equipped with multimedia equipment that is connected to the corporate network DGINH and integrated in the educational process for use existing digital multi-media educational resources of the institute with access to the Internet.

Our students actively participate in the annual Internet-based testing on various educational programs.

“Institute of Entrepreneurship Cisco”, functioning under the “Technologies of successful entrepreneurship in the North Caucasus Federal District of Russia”, is operating at the Institute.

This program is supported by U.S. Agency for International Development, USAID, an international corporation Cisco Systems, Inc., office of the non-profit organization "Project Harmony" (USA) in Russia.

Lectures, conducted in the multimedia classrooms of the Institute, are read by visiting professors from leading universities of Russia, Germany, Slovakia and Great Britain with the use of modern telecommunication technologies. That has become the norm at the Institute.

Introduction of web-telecommunication systems helps to further development of our university. This solves such problems as the widespread introduction of distance learning; organizing and controlling educational and business processes; further development of the network of educational and informational courses and materials with animation; complex support of communication between students, faculty and staff.

The practice of recent decades and the applied innovations clearly show that there has been a dramatic breakthrough in the implementation of new ideas and teaching methods in the education system in many countries. New approaches and ideas will allow organizing educational process at a new qualitative and technological level.

Composition-Nominative Transition and Temporal Logics of Functional-Equational Level

Oksana Shkilnyak
Department of Information Systems
Taras Shevchenko National University of Kyiv,
64, Volodymyrska Street, 01033
Kyiv, Ukraine
me.oksana@gmail.com

Abstract—**Composition-nominative transition logics (CNTL) are special program-oriented logics of partial predicates based on composition-nominative logics of quasi-ary predicates and traditional modal logics. In the paper first-order CNTL of functional-equational level are proposed. Composition-nominative modal transition system (MTS) is a basic concept of CNTL. We define various variants of general and temporal MTS, in particular, functionally stable MTS. Logics, based on such systems, are specified. Sequent calculi are constructed for the defined logics and soundness and completeness theorems are proved.**

Composition-nominative logics, modal logics, temporal logics, sequent calculi, soundness, completeness (key words).

I. Introduction

Modal and temporal logics are successfully used for information and program systems description. Composition-nominative modal logics (CNML) [1] combine facilities of traditional modal logics and composition-nominative logics of quasi-ary predicates [2]. Composition-nominative logics are based on the common for logic and programming composition-nominative approach [3] which is grounded on the following main principles: principle of ascending from the abstract to the concrete, principle of subordination of syntax to semantics, compositionality principle and nominativity principle.

Composition-nominative modal system (CNMS) [1] is the central concept of CNML. Considering the fact of changing and development in subject-domains, transition CNML based on modal transition systems (MTS) – a variant of CNMS – were specified to describe transitions from one state of the universe to another. Such traditional logics as alethic, temporal, epistemic etc can be considered within the scope of MTS. We defined general and temporal modal transition systems as important cases of MTS. Basing on special refinement of the notion of composition-nominative modal system, new classes of transition and temporal CNML of renominative and first-order levels were defined and investigated in [4-6]; for these logics sequent calculi were also constructed.

In this paper first-order CNML of functional-equational level are studied. We specify semantic models and languages for the introduced logics and investigate their semantic properties. We define functionally stable MTS and construct sequent calculi for functionally stable general and temporal MTS.

II. Composition-Nominative Modal Transition Systems

Composition-nominative modal transition system (MTS) is a basic concept of CNTL. We define MTS of functional-equational level as an object $\boldsymbol{M} = ((S, \boldsymbol{R}, Fn \cup Pr, C), Tr \cup Fm, I)$, where

- S is a set of states of the universe,
- $\boldsymbol{R}$ is a set of relations $R \subseteq S \times S$,
- $Fn \cup Pr$ is a set of functions and predicates on states,
- C is a set of compositions on $Fn \cup Pr$,
- $Tr \cup Fm$ is a set of terms and formulas of the modal language,
- I is an interpretation mapping of terms and formulas over states of the universe.

A set of compositions of CNTL is determined by basic logical compositions of functional-equational level and basic modal compositions.

General MTS (GMTS) and temporal MTS (TMTS) are important variants of MTS. For these MTS $\boldsymbol{R}$ consists of a unique relation $\triangleright$. Modal composition $\Box$ (necessarily) is a basic modal composition for GMTS. For TMTS, we take modal compositions $\Box\uparrow$ (it will always be) and $\Box\downarrow$ (it was always) as basic modal compositions.

Modal composition $\Diamond$ (possibly) can be expressed by $\Box$: $\Diamond P$ stand for $\neg\Box\neg P$.

Similarly, modal compositions $\Diamond\uparrow$ (it will sometime be), and $\Diamond\downarrow$ (it was sometime) are derived from $\Box\uparrow$ and $\Box\downarrow$: $\Diamond\uparrow P$ and $\Diamond\downarrow P$ stands for $\neg\Box\uparrow\neg P$ and $\neg\Box\downarrow\neg P$ correspondingly.

Depending on restrictions on the relation $\triangleright$, various variants of GMTS and TMTS can be defined. For example, let us consider reflexivity, transitivity, and symmetry. If $\triangleright$ is

reflexive, we add an *R*-prefix to the name of the MTS, for transitivity we write *T*, and *S* means symmetry. Thus, we can get the following systems of GMTS:

R-GMTS, *T*-GMTS, *S*-GMTS, *RT*-GMTS, *RS*-GMTS, *TS*-GMTS, *RTS*-GMTS.

By analogy, similar systems can be obtained for TMTS.

Basic logical compositions of functional-equational level are [2] $\neg$ (negation), $\vee$ (disjunction), $\exists x$ (existential quantifier), $S^{\bar{v}}$ (superposition), $'x$ (denomination), $=$ (equation).

Quantifier compositions $\forall x$ is a derived one: $\forall xP$ stands for $\neg\exists x\neg P$.

For MTS of functional-equational level, let us specify a set of states of the universe *S* as a set of neoclassical [2] structures $\alpha=(A_\alpha, Fn_\alpha\cup Pr_\alpha)$, where Fn_α and Pr_α are sets of *V*-quasi-ary equitone functions ${}^{V}A_\alpha\to A_\alpha$ and *V*-quasi-ary equitone predicates ${}^{V}A_\alpha\to\{T,F\}$ correspondingly. Thus, $Fn=\bigcup_{\alpha\in S} Fn_\alpha$ and $Pr=\bigcup_{\alpha\in S} Pr_\alpha$ are sets of functions and predicates over data of all the states of the universe, $A=\bigcup_{\alpha\in S} A_\alpha$ is a set of basic data of the universe.

Language of functional-equational MTS. An alphabet of MTS of functional-equational level consists of a set of individual names *V*, sets of denomination symbols *Dns*, functional symbols *Fns*, predicate symbols *Ps*, symbols of basic compositions $\neg$, $\vee$, $\exists x$, $S^{\bar{v}}$, $'x$, $=$ and a set of basic modal compositions *Ms* (modal signature of TMS.

For GMTS, $Ms=\{\Box\}$; for TMTS, $Ms=\{\Box\uparrow, \Box\downarrow\}$.

Sets of terms *Tr* and sets of formulas *Fm* are inductively defined as follows.

T1. Each functional symbol is an (*atomic*) term; each denomination symbol is an (*atomic*) term.

T2. Let $t, t_1,..., t_n$ are terms. Then $S^{v_1,...,v_n} t\, t_1...t_n$ is a term.

F1. Every $p\in Ps$ is an (*atomic*) formula.

F2. If Φ and Ψ are formulas, then $\neg\Phi$ and $\vee\Phi\Psi$ are formulas.

F3. If Φ is a formula, then $S^{v_1,...,v_n}\Phi\, t_1...t_n$ is a formula.

F4. If Φ is a formula, then $\exists x\Phi$ is a formula.

F5. Let t and s are terms. Then $=ts$ is a formula.

M. If Φ is a formula and $\mathfrak{K}\in Ms$, then $\mathfrak{K}\Phi$ is a formula.

For GMTS and TMTS, the rule M above is specified by rules MG and MT correspondingly:

MG. If Φ is a formula, then $\Box\Phi$ is a formula.

MT. If Φ is a formula, then $\Box\uparrow\Phi$ and $\Box\downarrow\Phi$ are formulas.

The type of the functional-equational TMS is specified by its modal signature *Ms*, similarity of transition relation *R* for each $\mathfrak{K}\in Ms$, and signature of insignificance [1, 2].

Let us define an interpretation mapping of terms and formulas on states. Firstly, we specify such mapping for atomic terms and formulas: $I: Fs\cup Ps\times\boldsymbol{S}\to Fn\cup Pr$. For each $f\in Fs$ and $p\in Ps$ we have $I(f,\alpha)\in Fn_\alpha$ and $I(p,\alpha)\in Pr_\alpha$, and also $I('v)='\boldsymbol{v}$ for every $'v\in Dns$. Then, we continue the defined mapping to $I: Tr\cup Fm\times\boldsymbol{S}\to Fn\cup Pr$ by the following rules.

IT. $I(S^{v_1,...,v_n} t\, t_1...t_n,\alpha)=S^{v_1,...,v_n}(I(t,\alpha), I(t_1,\alpha),..., I(t_n,\alpha))$.

IF1. $I(\neg,\alpha)=\neg(I(\Phi,\alpha))$; $I(\vee\Phi\Psi,\alpha)=\vee(I(\Phi,\alpha), I(\Psi,\alpha))$.

IF2. $I(S^{v_1,...,v_n}\Phi\, t_1...t_n,\alpha) = S^{v_1,...,v_n}(I(\Phi,\alpha), I(t_1,\alpha),..., I(t_n,\alpha))$. $\boldsymbol{I}(=ts\,\alpha)==(I(t\,\alpha), I(s\,\alpha))$.

IF3. $I(\exists x\Phi,\alpha)(d) =$

$$=\begin{cases} T, & \text{if there is } a\in A_\alpha \text{ such that } Jm(\Phi,\alpha)(d\nabla x\mapsto a)=T,\\ F, & \text{if } Jm(\Phi,\alpha)(d\nabla x\mapsto a)=F \text{ for all } a\in A_\alpha,\\ & \text{else undefined.}\end{cases}$$

For GMTS, we use:

IMG. $I(\Box\Phi,\alpha)(d) =$

$$=\begin{cases} T, & \text{if } Jm(\Phi,\delta)(d)=T \text{ for all } \delta\in S \text{ such that } \alpha\triangleright\delta,\\ F, & \text{if there is } \delta\in S \text{ such that } \alpha\triangleright\delta \text{ and } Jm(\Phi,\delta)(d)=F,\\ & \text{else undefined.}\end{cases}$$

For TMTS, we add:

IMT. $I(\Box\uparrow\Phi,\alpha)(d) =$

$$=\begin{cases} T, & \text{if } Jm(\Phi,\delta)(d)=T \text{ for all } \delta\in S \text{ such that } \alpha\triangleright\delta,\\ F, & \text{if there is } \delta\in S \text{ such that } \alpha\triangleright\delta \text{ and } Jm(\Phi,\delta)(d)=F,\\ & \text{else undefined.}\end{cases}$$

$I(\Box\downarrow\Phi,\alpha)(d) =$

$$=\begin{cases} T, & \text{if } Jm(\Phi,\delta)(d)=T \text{ for all } \delta\in S \text{ such that } \delta\triangleright\alpha,\\ F, & \text{if there is } \delta\in S \text{ such that } \delta\triangleright\alpha \text{ and } Jm(\Phi,\delta)(d)=F,\\ & \text{else undefined.}\end{cases}$$

Here $I(t,\alpha)\in Fn_\alpha$ and $I(\Phi,\alpha)\in Pr_\alpha$.

Let t_α and Φ_α stand for a function $I(t,\alpha)$ and a predicate $I(\Phi,\alpha)$, which is a value of t and Φ on a state α.

Φ is true for the MTS *M*, if Φ_α is true predicate for each $\alpha\in S$. It is denoted by $M\models\Phi$.

Φ is logically valid (notation: $\models\Phi$), if $M\models\Phi$ for all MTS *M* of the same type.

We will use more usual notation $t=s$ instead of $=ts$.

III. Semantic Properties and Sequent Calculi of Functional-Equational MTS

Let $d\notin{}^{V}A_\delta$. Then, let us assume that $\Phi_\delta(d)=\Phi_\delta(d\cap{}^{V}A_\delta)$ (here $d\cap{}^{V}A_\delta$ is a shortened notation for the name set $[v\mapsto a\in d\,|\,a\in A_\delta]$). Let us call it general condition of determinacy on states, and MTS with such condition a *Gn*-MTS [5]. Further only this type of MTS will be considered.

Theorem 1. Basic compositions of functional-equational Gn-MTS preserve equitonicity.

Semantic properties of functional-equational MTS. Let us study specific properties of functional-equational MTS for modalities, superpositions and equality.

Theorem 2. 1. Formulas $\Box t=s\to\Box\ S^{x,\bar{v}}(\tau,t,\bar{t}) = S^{x,\bar{v}}(\tau,s,\bar{t})$ are logically valid.

2. Formulas $\Box t=s\to\Box\ S^{x,\bar{v}}(\Phi,t,\bar{t})\leftrightarrow S^{x,\bar{v}}(\Phi,s,\bar{t})$ are logically valid.

At the same time, we have the following examples [6].

Example 1. Formulas $t=s \to \Box\, t=s$ та $\Box\, t=s \to t=s$ are not logically valid.

Example 2. Formulas $\Box\ S^{\bar{x}}(\Phi,\bar{t}) \to S^{\bar{x}}(\Box\Phi, \bar{t})$ та $S^{\bar{x}}(\Box\Phi,\bar{t}) \to \Box\, S^{\bar{x}}(\Phi,\bar{t})$ are not logically valid.

Thus, for functional-equational MTS, superpositions can not be carried over modalities. The reason is that same functional symbols can be differently interpreted on different states. However, denomination symbols are interpreted identically on different states of various modal systems.

GMTS on which formula $\forall x \Box\Phi \to \Box\, S^{\bar{x}}(\Phi,\bar{t})$ is refuted was constructed in [6]. For traditional modal logic, relational model are known on which $\forall x \Box\Phi(x) \to \Box\Phi_x[t]$ is refuted.

We can avoid the impossibility of carrying superpositions over modalities at the cost of preservation of values of basic functions while moving to subsequent states. Let us call this condition a functional stability:

MTS M is *functionally stable*, if for all $\alpha,\beta \in \boldsymbol{S}$ such that $\alpha \triangleright \beta$, for all $f \in Fns$, $d \in {}^{V}A_{\alpha}$ and $f_{\alpha}(d)\downarrow$ and $f_{\beta}(d \cap {}^{V}A_{\beta})\downarrow$ hold, then we have $f_{\alpha}(d) = f_{\beta}(d \cap {}^{V}A_{\beta})$.

Theorem 3. If MTS M is functionally stable, then for all $\alpha,\beta \in \boldsymbol{S}$ such that $\alpha \triangleright \beta$, for all $\tau \in Tr$ and $d \in {}^{V}A_{\alpha}$, and if $\tau_{\alpha}(d)\downarrow$ and $\tau_{\beta}(d \cap {}^{V}A_{\beta})\downarrow$ hold, then $\tau_{\alpha}(d) = \tau_{\beta}(d \cap {}^{V}A_{\beta})$ holds.

The condition of functional stability is necessary and sufficient for carrying superpositions over modalities.

Theorem 4. GMTS M is functionally stable $\Leftrightarrow$ $M \models \Box\, S^{\bar{x}}(\Phi,\bar{t}) \leftrightarrow S^{\bar{x}}(\Box\Phi,\bar{t})$.

Theorem 5. For functionally stable GMTS $M \models t=s \leftrightarrow \Box\, t=s$ holds.

By analogy, statements similar to theorems 2, 4 and 5 can be obtained for TMTS.

Let us introduce a relation of logical consequence for sets of specified with state names formulas.

Δ is a logical consequence of Γ in MTS M (notation $\Gamma \models_M \Delta$), if for all $d \in {}^{V}A$, $\Phi_{\alpha}(d \cap {}^{V}A_{\alpha}) = T$ for all $\Phi^{\alpha} \in \Gamma$ implies the impossibility that $\Psi_{\beta}(d \cap {}^{V}A_{\beta}) = F$ for all $\Psi^{\beta} \in \Delta$.

Δ is a logical consequence of Γ in MTS of a certain type (notation $\Gamma \models \Delta$),), if $\Gamma \models_M \Delta$ for all MTS M of the same type.

Modalless properties of the defined logical consequence are identical to the corresponding properties for composition-nominative logics of quasi-ary predicates [2].

Then, for functionally stable GMTS we have:

$S\Box\vdash)$ $\boldsymbol{\Gamma}, S^{\bar{x}}(\Box\Phi,\bar{t})^{\alpha} \models_{\boldsymbol{M}} \boldsymbol{\Delta} \Leftrightarrow \boldsymbol{\Gamma}, \Box\, S^{\bar{x}}(\Phi,\bar{t})^{\alpha} \models_{\boldsymbol{M}} \boldsymbol{\Delta}$.

$S\Box\dashv)$ $\boldsymbol{\Gamma} \models_{\boldsymbol{M}} \boldsymbol{\Delta}, S^{\bar{x}}(\Box\Phi,\bar{t})^{\alpha} \Leftrightarrow \boldsymbol{\Gamma} \models_{\boldsymbol{M}} \boldsymbol{\Delta}, \Box\, S^{\bar{x}}(\Phi,\bar{t})^{\alpha}$.

Modal properties of functional-equational MTS are defined the same way for propositional and nominative levels [4]. For GMTS we have:

$\Box\vdash)$ $\Box\Phi^{\alpha}, \Gamma \models_M \Delta \Leftrightarrow \{\Phi^{\beta} \mid \alpha \triangleright \beta\} \cup \Gamma \models_M \Delta$.

$\Box\dashv)$ $\Gamma \models_M \Delta, \Box\Phi^{\alpha} \Leftrightarrow \Gamma \models_M \Delta, \Phi^{\beta}$, for a state $\beta \in S$ such that $\alpha \triangleright \beta$.

For TMTS, we get properties for the future and the past: $S\Box\uparrow\vdash$, $S\Box\downarrow\vdash$, $S\Box\uparrow\dashv$, $S\Box\downarrow\dashv$ and $\Box\uparrow\vdash$, $\Box\downarrow\vdash$, $\Box\uparrow\dashv$, $\Box\downarrow$ instead of $S\Box\vdash$, $S\Box\dashv$ and $\Box\vdash$, $\Box\dashv$.

Sequent calculi for functionally stable GMTS and TMTS. Sequent calculi for MTS are constructed basing on their relational semantics. Let us call prefixes $\alpha\vdash$ and $\alpha\dashv$ state specifications. Here α is a state on which a specified formula is considered. Sequents are enriched with constructed on the current derivation stage sets S and R. Enriched sequent: $\Sigma \,//\, St \,//\, M$, where St is constructed on the current derivation stage set of state names with sets of their basic data. These sequents allow to extend state names carriers. Such sequents looks like $\Sigma \,//\, \alpha\{A_{\alpha}\}, \ldots \,//\, M$, where $\alpha\{A_{\alpha}\}, \ldots$ are constructed on the current derivation stage states with sets of their basic data and M is a scheme of the model of the universe.

Non-modalised forms don't affect state specification prefixes and a scheme of a model of the universe. Such properties derive from calculi of logic of quasi-ary predicates [4]. The forms of functional-equational level: $\vdash\neg$, $\dashv\neg$, $\vdash\vee$, $\dashv\vee$, $\vdash SS\Phi$, $\dashv SS\Phi$, $\vdash S\neg$, $\dashv S\neg$, $\vdash S\vee$, $\dashv S\vee$, $\vdash S\exists$, $\dashv S\exists$, $\vdash S\exists\exists$, $\dashv S\exists\exists$, $\vdash\exists$, $\dashv\exists$,$\vdash ZN\Phi$, $\dashv ZN\Phi$, $\vdash Z\Phi$,$\dashv Z\Phi$, $\vdash SE$, $\dashv SE$. Auxiliary forms $\vdash TrN$ and $\dashv TrN$ are used for normalization of terms. For axioms of symmetry and transitivity, we use auxiliary forms ESm and ETr. Axiom of reflexivity is taken into account by adding the condition of sequent closure: a presence of a formula $\alpha\dashv t=t$. Forms $\vdash S\Box$ and $\dashv S\Box$ are used for carrying a modality over a superposition; this rule is correct as long as the condition of functional stability holds.

Sequent forms for modal compositions are similar to the corresponding forms for propositional and quantifier levels (see [4]). The shape of these forms depends on properties of a reachability relation $\triangleright$ on states of the universe. We have the forms $\vdash\Box$ and $\dashv\Box$ for GMTS, and $\vdash\Box\uparrow$ $\vdash\Box\downarrow$ and $\dashv\Box\uparrow$, $\dashv\Box\downarrow$ for TMTS.

Theorem 6 (soundness). Let sequent $\vdash\Gamma\dashv\Delta$ is derivable. Then $\Gamma \models \Delta$.

We use the method of Hintikka's model set systems [7] for proving the completeness theorem for sequent calculi of functional-equational GMTS and TMTS.

Theorem 7 (completeness). Let $\Gamma \models \Delta$. Then sequent $\vdash\Gamma\dashv\Delta$ is derivable.

IV. Conclusion

Composition nominative modal and temporal logics of functional-equational level are studied in this paper. We specify semantic models and languages of such logics and study their semantic properties. Interaction between superpositions, quantifiers and modal compositions is investigated; we define functionally stable modal systems which allow of carrying modalities over superposition. Basing on properties of relation of logical consequence for sets of formulas, sequent calculi are constructed for general transition and temporal logics of functional-equational level. The soundness and completeness theorems hold for these calculi.

REFERENCES

[1] M. Nikitchenko and S. Shkilnyak, "Composition-Nominative Modal Logics," Problems in Programming, vol. 1-2, 2002, pp. 27-33. (in Ukrainian)

[2] M. Nikitchenko and S. Shkilnyak, Mathematical Logic and Theory of Algorithms. Publishing house of National Taras Shevchenko University of Kyiv, 2008, 528 p. (in Ukrainian)

[3] M. Nikitchenko, "Composition-nominative approach to the refinement of the concept of a program," Problems in Programming, vol. 1, 1999, pp. 16-31. (in Russian)

[4] O. Shkilnyak, "Composition-Nominative Modal and Temporal Logics: Semantic Properties and Sequent Calculi," Scientific Notes of NaUKMA, vol. 86: Computer science series, 2008, pp. 25-34. (in Ukrainian)

[5] O. Shkilnyak, "Semantic Properties of Composition-Nominative Modal logics," Problems in Programming, vol. 4, 2009, pp. 11-23. (in Ukrainian)

[6] O. Shkilnyak, "Composition-Nominative Modal Logics of Functional-Equational Level," Problems in Programming, vol. 2-3, 2010, pp. 42-47. (in Ukrainian)

[7] V. Smirnov (ed.), Semantics of Modal and Intentional Logics. Progress, Moscow, 1981, 494 p. (in Russian)

Future Outlook and Opportunities of Information Technology and Job Creation in IT-Sector of Republic of Dagestan

Yahya Buchaev, PhD, Professor, Rector,
Vladimir Galyaev, Ph.D., Head of the Department of Information Technology
Karahan Radjabov , Ph.D., associate professor, Dean of the faculty of "Applied Informatics"
Makhachkala, Dagestan State Institute of National Economy under the Government of Republic of Dagestan (DGINH), dginh@yandex.ru

Abstract – The article deals with the problems of development tendencies of the IT market and the opportunities of job creating in the economy sector in Dagestan.

Keywords- Dagestan, information technology, job creating, economy sector, tendencies, IT market, development prospects.

The development and wide application of information technology (hereinafter - IT) in all economic sectors of RD is a necessary trend in the development of our republic.

As international experience shows, the use of IT is critical in improving the living standards of citizens, national economic competitiveness, human capital development and modernization of government institutions.

The transformation of the IT industry into a driving force for economic growth and modernization of the region in the short term is possible only in case of providing targeted support from the government of the Republic.

Under the IT market we mean the set of such segments as:

- Production and sale of computer equipment;
- Developing and selling software products;
- Provision of services related to the implementation and maintenance of IT;
- Provision of telecommunication services.

In the following discussion under the IT industry we will mean manufacturing products by Russian companies and providing services in these segments.

Decrease in the proportion of sales of computer hardware and accessories in the total market volume and sales growth of services in IT and software development can be attributed to market trends of IT.

The important phenomenon in the IT market is that production is moving from developed countries to the countries, characterized by a low cost of wages and favorable taxation, which is especially important for development of software that does not require the presence of a complex infrastructure.

Developed telecommunication infrastructure is a prerequisite for large producers. The market of IT-services in 2008-2009 was to a large extent influenced by the crisis, but compared to other sectors of the economy, it looked good. In this segment, the decline in turnover in 2009 amounted to 25-30%.

In 2008, experts estimated the Russian market of IT-services and system integration of 5.5 billion dollars; in 2009 this figure had dropped to $ 4 billion, and in 2010 it amounted to $ 4.1 billion. Forecast for 2011 is 4.9 billion dollars.

Russia IT market is only 1.5% of the total Russian GDP. For comparison, the U.S. IT market exceeds $ 500 billion (more than 5% of GDP). According to CNews, market of IT-services in Sweden is - 9, Spain - 15, Germany - 43 billion dollars.

According to the report of CNews100, revenue of hundred largest IT companies in Russia in 2010 amounted to about 757 billion rubles. Growth was 45% compared to 2009 data. The most successful, according to analysts, were the segments of IT - services and software.

Currently, only 14 percent of the Russian IT market comes from exports. For comparison, in other countries with fast growing IT industries, the export share is predominant in the overall structure and is, for example, 70% in Israel and 80% in India.

The development of IT industry in Russia and in Dagestan, in particular, is constrained by a number of barriers listed in the Concept of Development of the IT market in Russia to 2010.

Fairly big problem is the technical nature of the lack of affordable high-speed Internet in the territory of Dagestan, which seriously hinders the development of the industry.

The low level of proficiency in English among most of the professionals in the IT-market is also quite a strong deterrent.

Underdevelopment and poor availability of telecommunication infrastructure hinders the development of small and medium enterprises sector, thereby preventing their entry to the world market and impeding development of relations with foreign partners.

Discrepancy between the system of professional training of IT specialists in Russia and the world's leading training standards leads to a shortage of staff with relevant expertise, especially mid-level professionals, programmers and project managers of information, and makes it hard to compete effectively with foreign experts.

IT companies alone cannot solve these problems. For this purpose, it is necessary to carry out coordinated regional policy aimed at removing these barriers, and provide state support for the IT market in Russia and Dagestan.

Currently software companies in Dagestan have no wish to take orders from local organizations, since the prices for these services in Dagestan are substantially lower than in Russia. Known firms of Republic (Bevolex, Color IT, ICT, etc.) are engaged in projects that are ordered from the central regions.

Accordingly, IT firms do not promote their services to develop software for the domestic market. Therefore many companies, not knowing that we have our own software developers, are forced to turn outside of Dagestan, which leads to an outflow of funds.

The most of companies mainly focus on development and support of sites. Both proven and young firms are involved in this activity as well as single developers. However, this is a very narrow area in the IT-sphere.

Mainly young people (often graduate students) are on the staff of most computer companies. With experience in such work further on they get job in a more reputable company or government organization, or go beyond the limits of Dagestan. Thus, most computer companies faced the challenge of staff turnover.

An important aspect for IT-industry is to improve English language teaching in educational institutions of Republic, since mastering English is the main competitive advantage for the training of highly qualified IT-specialists.

Accelerated formation of specialized industrial parks and a modern telecommunication infrastructure in the regions is necessary to ensure the active development of industry in Russia. The system of professional training must be improved, financial resources should be available, and more effective mechanisms of protecting intellectual property must be provided.

One way out of this situation would be to create specialized centers for the developement and implementation of software at the leading universities of the country. To do this, there are some positive assumptions:

1. The availability of the necessary material base: production facilities, telecommunications, hardware and software, that removes a number of cost for the organizing of such centers;

2. The selection of the best-prepared to work in the field of programming specialists from senior students and graduates;

3. A large number of applied problems in the field of economy and education, which demands solutions. That will provide future orders for these centers;

4. Chance to recharge the staff through continuous student engagement in the projects (3-4 course - trainee, 4-5 course - employee).

However, in order to solve a series of questions that cannot be solved at higher schools, the government of the Republic should provide support for implementing these ideas:

1. Purchase of licensed development tools, which will pay for itself in the case of smooth operation of the centers;

2. Providing specialized computer equipment (This problem is partially solved by the universities with the help of planned purchases of equipment);

3. Providing specialized centers with orders.

At this stage of development of the country there are already a lot of such problems. Budgets of all ministries, departments and agencies provide articles for the IT needs. Software used at different levels of state and municipal government is obsolete, and its purchase from specialized firms outside of Dagestan is rather costly. This can be done with a tool of state orders. To make the process transparent, tenders can be declared specifically for software development centers.

Further development of the IT industry is seen as expanding the network of companies-developers. It will happen at the expense of the graduates of these centers, which ultimately will be able to start their own business in IT sphere.

This will provide new jobs for the republic, increase the investment attractiveness and help create healthy competition in the IT-sector.

In the first phase of development of this sector it would seem that the invested funds will be wasted, i.e., we will not see any significant return on investment.

But, in our opinion, the creation of a favorable environment will attract enterprising IT-specialists, subject to availability of funding; that is, demand will shape the proposal.

Logical chain - setting goals and providing financial assistance in solving them; creating incentives for opening new IT - companies; the opportunity to get a good IT - education with the prospect of well-paying job - will lead to the development of an information cluster in Dagestan, which can provide a sufficiently large number of new jobs.

Improving the system of vocational training for the IT industry and its compliance with basic international standards is the pressing factor for the development of the industry.

Measures aimed at improving the education system will help solve the problem of providing industry with qualified personnel in the long run. We believe that it is necessary to ensure the adaptation of higher education to the needs of the IT industry, to bring educational programs to train IT specialists in line with international standards, as well as to organize work at the courses for advanced training in specialties that are in demand in IT industry.

To solve this problem it is necessary to ensure the involvement of representatives of leading Russian companies engaged in the sphere of IT industry, in developing programs and organizing practical training in this area (teaching and practice centers).

It is necessary to provide a system of state statistical observation of the development of IT-market of the republic. This will allow adequately assess the dynamics of its growth, which would increase the tax base.

As a result of the introduction of IT projects in other industries, labor productivity will increase, which would also increase budget revenues.

REFERENCES

[1] The Concept of Development of the IT market in Russia to 2010.

[2] Regions of Russia. Socio-economic indicators for 2010. Statistical handbook of the Federal State Statistics Service. - M.: 2010.

Improvement of Administrative Procedure business processes running within regional self-governmental institutions

Michal Grell
Department of Applied Informatics
Faculty of Economic Informatics
University of Economics in Bratislava
Dolnozemská cesta 1, 852 35 Bratislava
Slovakia
grell@euba.sk

Zuzana Mikitová, Gabriela Rotterová
students of engineering study: Managerial Decision Making and Information technology
Department of Applied Informatics
Faculty of Economic Informatics
University of Economics in Bratislava
Dolnozemská cesta 1, 852 35 Bratislava
Slovakia
mikitova.zuzana7@gmail.com, rotterova@zmail.sk

Abstract- **The paper deals with analysis of problems closely related to improvement of quality and efficiency within functionality of services provided by regional self-governmental institutions. However, the paper also deals with administrative procedure business process modeling and their improvement within business process management as well. On the other hand, a set of possibilities related to interconnected on-line communication among governmental and self-governmental information systems via public administration portal are discussed in the paper too.**

Keywords-local government; modeling; New Public Management; Balanced Scorecard; Business Process Management; Business Activity Monitoring; administrative procedure; public administration portal;

I. INTRODUCTION

An administrative procedure or action represents a legal process, where an appropriate administrative body makes decisions related to rights and duties of natural or legal personalities, while an administrative body and administrative procedure actors are considered to be the procedure participants. A creation of an appropriate certificate or acknowledgement or decay of a particular right or duty may be a result of the above-mentioned procedure. An electronic service utilizing the village or town data center in order to provide support in digitizing of administrative procedure processes and to prepare conditions for handling or arrangement of the actual submitting received by an appropriate self-governmental institution in a very short time, is considered to be an improvement of administrative procedure processes. However, an adequate transformation of information technology systems, which created an integral part of an appropriate self-governmental institution related to this type of services is required and supposed as well With respect to these aspects, we are dealing with modeling of administrative procedure business processes running within self-governmental institutions too.

II. TRANSITION FROM FUNCTION TO PROCESS ORIENTED MANAGEMENT

BUSINESS PROCESS MANAGEMENT TOOLS

The aim of the territorial self-government concept [4] is to create a set of conditions for implementation and operation of information technology systems, so that a quality and efficiency related to functionality and services provided by self-governmental institutions could be improved and on-line communication among self-governmental and governmental institutions might be achieved. The conditions based on process-oriented management and creation of integrated territorial self-government environment enables realizing these key activities in a great deal.

At present, an implementation of several methods applied within commercial firms, companies and institutions into governmental and self-governmental institutions is considered to be a trend, which represents a transition from function to process oriented management. However, a set of new approaches and methods denoted as New Public Management (NPM System) and based on a model closely related to managerial procedures and tools applied within private firms and companies are being transferred and implemented in public sector as well [5]. The New Public Management is applied and oriented to the following regional self-government areas:

- Assurance of citizen content via services provided for them by self-governmental institutions acting in the town or village.
- Improvement of efficiency related to self-governmental institutions acting in the town or village.
- Savings in the town or village budget.

A clear identification of business process, which also generates an appropriate set of outputs within self-governmental institutions, is considered to be the principal basis for implementation of NPM System. On the other hand, the *Business Process Management (BPM)* seems to be an adequate solution, while this system consists of activity sets, which enable changing an organization management oriented to business processes instead of traditional functional hierarchy.

Application of BPM method in design of optimized business processes creates the principal basis for support of management with the use of the NPN Standards, while maximum content of the citizen together with maximum efficiency in doing appropriate activities, plays a role of great importance. However, the Balanced Scorecard Method (BSC Method) application is recommended for these purposes as well, while this method is considered to be an important tool for strategic management of regions incl. cities, towns and villages and is presented as a real support tool, when preparing a management concept and implementing strategy closely related to regional self-government development. The BSC Method is applied for public sector management in abroad in a great deal, however it is not applied within public sector management in the Slovak republic very much. There is an important aspect, which indicates a set of differences in BSC perspective hierarchies, when considering private and public sector firms, companies and institutions [7]. The final business goals and aims postulated within private sector firms, companies and institutions are closely related to the Financial Perspective, while for the public sector firms, companies and institutions, the principle aims and goals of business are closely related to Customer (Citizen) Perspective. The BSC model is quite different for private sector firms, companies and institutions and for public sector ones, as a result of that, while it respects appropriate differences in mission statement, objectives and activities of both sectors.

The above-mentioned approaches are being implemented within e-Government system, which represents transformation of self-governmental internal and external relations with the use of information technology systems, while the aim is to optimize internal and external business processes as well. The legal determination of rules, which create conditions for clear definition of business processes running within governmental self-governmental firms, companies is considered to be the principal issue related to e-Government task fulfillment.

Most of commercial BPM applications and tools enable a user-friendly creation of simple user-oriented forms and screens, which may be integrated with modern eForm solutions and with tasks closely related to the business processes to be modeled as well. On the other hand, the Business Activity and Step Monitoring (BAM) plays a role of great importance, while that monitoring enables discovering potential business process inefficiency.

III. MODELING OF SELECTED ADMINISTRATIVE PROCEDURE OBJECTS AND PROCESSES - THE MODELING AIM.

The administrative procedure is considered to be the most important governmental administrative business process. When modeling the administrative procedure, five principle processes shall be considered and respected: a) administrative procedure beginning, attestation of conformity, decision or verdict, decision or verdict survey or revision within appropriate appellate procedure and the decision or verdict execution.

At present, an enforced trend is closely related not only to information system design and implementation, however the modeling aim is to support these business process which have been created as a result of analyzed business process model as well, while the results create basis for integration of those business processes with adequate information technology systems. At present, this type of modeling becomes the principal basis of management and a relation between business processes and information system plays a role of greater and greater importance [3].

A creation of models related to administrative procedure objects and business processes is being done based on the conceptual level, where the content system model is being created and no aspects which deal with technology and implementation specification are not considered and respected. A set of different diagrams with specific abstraction and view angle level is applied, when designing the above-mentioned models, while several aspects are being stressed and some of are neglected as well. As a result of that, the diagrams which enable a different types of representation related to the modeled objects shall be applied, in order to get an entire view at the system to be modeled. The following diagrams shall be applied for the above-mentioned purposes: class diagram, state transition diagram and business process global model.

The class diagram contains these principle classes, which are considered to inevitable part of appropriate administrative procedure business processes. A visualization of relations among appropriate classes is considered to be the principal aim of modeling. Appropriate class attributes describe character of actual classes and the method names determine which actions will be done and what may achieve with the use of these methods. Subsequently, the described actions are applied when designing adequate state transition diagrams, while these diagrams represent the life cycles related to classes of objects.

Example: The class denoted as Request contains the following attributes: Request Number, Date of Acceptance, Date of Assignation, Refusal Date and Reason. However, this class contains a set of actions as well, while the set contains elements postulated as follows: Request Acceptance, Request

Processing, Propriety Check, etc. The association n: 1 exists between class denoted as Request and class denoted as Administrative Procedure Subject, while the Request creates basis for Administrative Procedure Subject as well.

We may have a detailed look at the Administrative Procedure process lifecycle with the use of State Transition Diagram. The class denoted as: Permit for specialized utilization of motorways, roads and local communications, which also an integral part of the Class Diagram was chosen for these purposes. The Transition State Diagram describes a set of states after having completed all actions related to an appropriate administrative procedure and concerned to an adequate object, from beginning to end of its life cycle.

The global model has been chosen in order to prepare a model concerned to the administrative procedure behavior, which seems to be most suitable for these purposes and is outgoing from individual administrative procedure business processes postulated as follows: administrative procedure beginning, attestation of conformity, decision, review of administrative action, administrative action execution. A set of appropriate administrative procedure business processes together with relating inputs and outputs are described via global model.

With respect the above-mentioned models, a need of process and object consistency is considered to be the principle issue for improvement related to analyzed processes, while that consistency is based on interaction of processes among each other and with adequate objects [3] and may be postulated as follows:

- Each class of objects from model of classes shall create an integral part of business process model, and shall be contained at least in one input, output or other external aspects related to the process model.
- Any business process input or output or any business process external aspect shall create an integral part of class model and play a role of class, association among classes or there may a combination of both as well.
- Any event, which is specified within transient description in the state transition diagram related to the class life cycle, shall correspond to that event specified within several process or processes.

It may be documented via Table 1, which implies the following statements:

TABLE I. EXAMPLE OF CONSISTENCY RELATED TO ANDMINISTRATIVE PROCEDURE BUSINESS PROCESSES AND OBJECTS

Class	Input	Output	Association	Event
Request	Request			
			Creates basis for	Review of request
Decision		Decision		
				Review of appeal
				Appeal decision - refusal

- The class denoted as *Request*, contained within class diagram, represents input for business process global model, where it create basis for process *Administrative procedure commencement*, simultaneously.
- The association denoted as *Creates basis*[1] represents an event denoted as *Review of request*, simultaneously.
- The class denoted as *Decision* from Class diagram is an output from process denoted as Decision within Global model of processes, simultaneously.
- The event denoted as *Review of appeal*, which is specified in the State Transition Diagram corresponds to the event *Appeal decision – refusal* in Global model of processes, simultaneously.

A set of consistency knowledge may be applied within regional self-government practice and for preparing of qualified proposals related to improvement of existing business processes interconnected to BPM as well.

IV. POSSIBILITIES OF PUBLIC ADMINISTRATION PORTAL FOR COMMUNICATION WITH CITIZENS VIA ELECTRONIC SERVICES

The existing information systems are being designed and implemented in form of isolated solutions, which do not respect a concept of information technology system and service oriented architecture implementation within public administration institutions, which should provide a clear management of internal business processes.

When considering electronic public administration maturity, we can say, that its infrastructure components are not built up completely. Only a few of services exist, which may be provided via appropriate information technology systems and a set of legal norms, standards a rules supporting electronic services within governmental and self-governmental institutions is very limited as well.

[1] This association has been described as the association between class denoted as *Request* and *Administrative procedure subject.*

Providing a set of complex and actual information is considered to be one form or possibility for supporting of citizens in applying information technology systems (e-Government), while their concentration in one location seems to be the best solution. The Central Public Administration Portal (hereinafter denoted as the GPAP System) should be a suitable solution in achievement this goal or aim, while it should provide access to public administration information resources and electronic services via WEB oriented technology as well. A level of provided electronic services is very different, while a transaction level is considered to be the highest one. It means, the citizen may gain forms or stationary documents for utilization of an appropriate public service via WEB Site; he/she may fill in that form and submit it via the same WEB Site subsequently.

An alignment of user to utilization of a concrete public administration electronic service with the use of relevant information resources is considered to be the most significant task of the GPAP System.

However, the GPAP System provides only a very limited set of transaction services as well. The citizens would be very happy, when they could utilize a transaction level of all provided services. They could save a lot of time, the business processes could be simplified and saving of travel costs together with no time limitation in disposal of an actual matter would play a role of great importance too.

An implementation of on line services within individual self-governmental institutions plays a role of great importance. On the other hand, we can say, that self-governmental institutions in the Slovak republic do not utilize the GPAP System services. Some of them have their own portals and utilize them in order to provide electronic services or they do not provide any electronic services at all.

The project CIVITAS started running in the Slovak republic. This project is considered to be innovative and enables providing electronic services for citizens and businessmen. The portal www.civitas.sk discovery is the first phase of the project. Thirteen towns and villages are members of this project. The citizen is allowed to send different submitting to institutions in the town of village, after having completed his/her registration and login.

The electronic services enable saving time and money for the citizens. Providing of electronic services (e-services) within more self-governmental institutions would be a great contribution for citizens. A utilization of the GPAP System and a connection to CIVITAS Project could be a suitable solution, with respect to financial aspects of portal implementation and operation as well.

The GPAP System enables providing a set of high-level information services. After having completed of the user's login he/she may utilize a set of further electronic services as well. The general submitting package is one of the above-mentioned services. This service becomes reachable since July 1, 2009. The General submitting service is applied for such submitting, where a content and form is not regulated by adequate legal norms and standards. Any text submitting may contain attachment, which may have a size limited to 14 Mbytes. You can send such submitting to the following institutions: Slovak Republic Government Office, The Top Level Check and Control Office of the Slovak republic, Attorney Generalship, all SR ministries, regional Land Register Offices, executor offices, and so on.

This electronic service offers a wide spectrum of possibilities and it could be very often utilized within portals and WEB Sites of different self-governmental institutions, where smaller towns and villages play a role of great importance. The self-governmental institutions in appropriate towns and villages could receive submitting sent by citizens and businessmen as well, while requests, complaints or announcements could represent such submitting types. Most of smaller towns and villages have not enough money for implementation and operation of their own portals and a utilization of the GPAP System could become a very suitable solution for them.

As a result of that, the citizen as a user of the town or village's WEB Site could find a reference to the GPAP System in the WEB Site main menu and after having completed his/her login; he/she could realize an appropriate submitting.

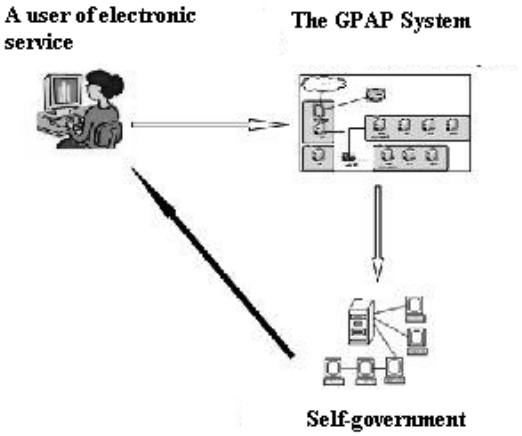


Fig.1. The submitting cycle

The citizen makes a selection of institution type and name and fills in the submitting body and sends the submitting within his/her next step. After having completed the above-mentioned procedure, the submitting life cycle is getting started and passing through the following phases (see also Fig.1):

- The user's registration in the GPAP System database

- After having completed his/her registration, the submitting is being sent to an appropriate local or municipal office.
- Subsequently, the submitting processing within appropriate local or municipal office is getting started.
- The citizen is informed about processing of his/her submitting

V. CONCLUSION

An improvement of business processes running within self-governmental institutions should be interconnected to e-Government implementation and operation. As a result of that, a direct check and control of the above-mentioned business processes by cities, towns and villages could be supported and an appropriate services interconnected to these services could be more accessible for citizens as well. A utilization of GPAP System at regional self-governmental level may be considered to be a good example in these branches. However, an implementation of business process management tools could generate a significant improvement, while a design and implementation of integrated town or village office together with data center for towns and villages could play a role of great importance.

REFERENCES

[1] M. Grell and I. Bandurič, Nová koncepcia riadenia v územnej samospráve. In *Ekonomické aspekty v územnej samospráve: recenzovaný zborník príspevkov z vedeckej korešpondenčnej konferencie* - Košice : Univerzita Pavla Jozefa Šafárika v Košiciach, 2011. ISBN 978-80-7097-863-4, s. 43-50. VEGA 1/0261/10.

[2] M. Kokles and M. Grell, Informačné systémy regiónov a verejnej správy. Bratislava: VŠEMvs, 2008. ISBN: 978-80-89143-71-9.

[3] V. Řepa, Podnikové procesy. Procesní řízení a modelování. 2. aktualizované a rozšířené vydání. Praha: Grada Publishing, 2007, s.177. ISBN 978-80-247-2252-8.

[4] Národná koncepcia informatizácie verejnej správy, schválená uznesením č. 331 vlády SR zo dňa 21.5.2008.

[5] M. Barzelay, *The New Public Management*. Improving Research and Policy Dialogue. University of California, 2001.

[6] P. Rumpel, *Městský marketing jako koncept rozvoje města*. In: Veřejná správa 29/2008, týdeník vlády ČR, 2008.

[7] Z. Hušek and M. Šusta and M. Půček, *Aplikace metody Balanced Scorecard (BSC) ve veřejném sektoru.* Výstup z projektu podpory jakosti č. 12/29/2006. Praha: Národní informační středisko pro podporu jakosti, 2006.

Socio-economic Aspects to Improve the Administration on the Basis of It-Technologies

Karahan Radjabov , Ph.D., associate professor,
Dean of the faculty of "Applied Informatics"
Makhachkala, Dagestan State Institute of National Economy under the Government of Republic of Dagestan (DGINH), dginh@yandex.ru

Abstract – The article deals with the problems of social and economic development of Dagestan Republic and influence It-techologies on this process.

Keywords - Dagestan, information technology, social and economic development, agriculture, prospects.

Russian state regional policy aims to ensure the leveling of socio-economic development in the constituent territories of the federation. The basic trends of this policy are:

- Creation of innovation centers of the economic growth in the light of competitive advantages of regions;
- Coordination of public and private investments to the infrastructure of regions which are targeted for economic growth;
- Alignment of the level and quality of life in the regions through the mechanism of social and fiscal policy.

The Development Strategy of the North Caucasus Federal District (N.C.F.D.) 2025 provides rapid development of region's economy, jobs creation and improvement of living standards.

The N.C.F.D. includes 7 regions of the Russian Federation. In 2010 the largest contribution (71%) to gross regional product (GRP) was made by Stavropol Region (36.6%) and the Republic of Dagestan (RD) (34.3%).

The analytical reference of the socio-economic development of Dagestan, prepared by the RF Ministry of Regional Development for 2008-2012, states that there has been a positive trend of socio – economic development indicators of the region.

The trend of constant increase of the index of gross capital formation is very encouraging. During the period from 2000 to 2010, this figure increased 21 times greater and it totaled 88122.5 mln., which indicates a high investment activity in the country.

The proportion of the rural population of Dagestan is 57.6% of the total population of the region, so the state and the level of agricultural development is crucial for socio-economic development. Agriculture of Dagestan specializes in these two sectors. Livestock specializes in sheep breeding to produce meat and wool. Crop production specializes in growing and processing of grains, vegetables, fruits and grapes. An index value agricultural output in Dagestan totaled 105.3% in 2010 compared with 2009. The trend analysis suggests that by 2011 the volume of agricultural output in Dagestan will be over 50 billion rubles. The proportion of Dagestan in Russia's total output of agricultural products was 1.8% (and 25.7% in the N.C.F.D.), which allowed to take 19th place in Russia's.

If we assess the structure of agricultural production in Dagestan from 2000 to 2010, we can make the following conclusions:

The volume of agricultural production in 2010 is the following: grain - 51.5%, potatoes - 97%, vegetables - 96.3%, cattle and poultry - 79.6%, milk - 81.9 %.

The farmers, including individual entrepreneurs, show good indicators of producing sunflower seeds (63.4%) and grain(17.4%), agricultural organizations in this respect seem to be unconvincing (grain - 31.1%, sunflower seeds - 29,1 %, potatoes - 0.3%, vegetables - 0.6%).

During the last 10 years, the percentage contribution of agricultural organizations to total production of grain decreased by half, while the farm population and individual farms have doubled the output.

The share of the certain categories of farms in total agricultural production in the first half of 2010 was characterized by the following data: agricultural

enterprises - 9.1%, peasant farms - 84.8%, private farms - 6.1%.

It is significant that in previous years in the lending process the majority of funds were allocated primarily to agricultural organizations, but the agro-industrial complex of the republic did not receive an adequate return. One of the tools for sustainable development of agriculture is to increase the availability of credit.

The credits worth 2773.1 million rubles were allocated for the "Development of agriculture and regulation of markets of agricultural products, raw materials and food for 2008-2012" program by the regional branch of "RosselkhozBank" PLC. Additionally, 340 million rubles have been allocated to support agro-industrial complex in Dagestan, 85.9% of the total comes from the federal budget, and 14.1% - from the republican budget.

On this basis, it is necessary to solve the problem of improving the investment climate in agro-industrial complex, making an emphasis on lending for investment projects of farmers who have made a decisive contribution to the development of the industry.

The analysis of economy and statistics shows that all branches of agriculture in Dagestan must solve the urgent task of improving management through the use of innovative tools and methods based on operational data processing on the status of this sector of the economy. It will provide the opportunities to identify the most important trends of development in a market economy, which are aimed at a qualitative breakthrough in the development of AIC of the region.

Among the most efficient forms of farming that can make a significant contribution to the development of agriculture. It is necessary to use modern information and communication technologies for the operational collection, storage, processing and subsequent analysis of the information about ongoing processes in the industry. All this will adopt a scientifically - based solutions to a wide range of such issues, as investment, subsidies, pricing, procurement, processing and marketing of agricultural products, strengthening the material and technical base.

Definite advance in this regard has already been made.

Electronic Document Management System, which operates in the Ministry of Agriculture, helps small forms of management in agriculture to receive well-coordinated information and advice, such as developing an accounting and tax accounting, legal consultation, and issues of regulation of land relations.

Large pool of opportunities and benefits is not used in the regional agribusiness in the absence of functional possibilities inherent in such data processing systems as the system of state information management in agriculture and Unified Information Support System, which were created in order to form government information resources and providing information for agricultural producers [1, 2].

Implementation of these systems at the regional level should help to provide services for agricultural producers and rural communities, to improve quality and management efficiency, to accelerate the pace of growth of agricultural production.

The program "The system of information on markets of agricultural product" was in great demand for the vast majority (87%) of respondents, as the sociological survey among managers of all categories of farms shows.

This interest was due to the fact that one of the major innovations in the development of automated information systems, attention has focused on the information needs of specific agricultural producers, in particular, to such information resources as:

- tariff rates, the volume of tariff concessions, the volume of import / export of basic agricultural products, raw materials and foods;
- indicators of the main types of agricultural products, raw materials and food in the whole country, and in some regions of Russia;
- prices for agricultural products, raw materials, food and industrial products;
- regulations, establishing the amount and procedure of realization of state support for agriculture, etc.

The system of market information is vital to agricultural regions. This is due to the fact that within its framework one can track prices and factors affecting prices for agricultural products, raw materials and food, material resources, fuel prices, etc.

Introduction of a Unified Information Support System will solve many problems:

- stable income as a result of economic activity of agribusiness;
- accurate and timely information on prices and markets for agricultural products;
- government support and subsidies for farmers;
- sound procurement planning of various types of agricultural products from abroad;
- participation in public procurement portal in real time, etc.;
- equalization of living standards of urban and rural population.

REFERENCES

[1] The system of public information support in the agriculture / Methodical information. - Moscow: Ministry of Agriculture, 2008, 108 p.

[2] K.J.Radjabov. Improving the management of agribusiness in the region with the use of information technology / III International Scientific and Practical Conference "Innovative Development of the Russian economy": v. 1. - Moscow, MESI, 2011, 240-242 p.

The Necessity of Process Improvement Municipalities

Peter Schmidt
Department of Applied Informatics
Faculty of Informatics
University of Economics
Bratislava, Slovakia
schmidt@atg-slovakia.sk

Jaroslav Kultan
Department of Applied Informatics
Faculty of Informatics
University of Economics
Bratislava, Slovakia
jkultan@gmail.com

***Abstract* - The paper deals with the necessity of improving document creation processes in municipalities with support for Web applications. Given the great diversity of communities in terms of population, social and demographic composition is very difficult to analyze certain processes in the management of the village. It is evident that the larger towns and more representative bodies, the process of creating official documents of the village much more difficult process than in a small village and a small Members' Forum. In general, the smaller the municipality, the less is the official documents. If the issue of support of local government documents, viewed from the perspective of using web applications get very sad information. These findings warranted to improve our processes in communities emphasize and encourage us to further work in analysis and design solutions to this problem.**

Keyword; process, local government, reengineering processes village website

I. INTRODUCTION

Document management is a fundamental activity local government offices. Their main task is to ensure timeliness and availability of these documents. In municipalities where there are established and agreed rules for the process of creating and managing internal documents of the village, as a rule generally binding regulations (hereinafter VZN) is a document management much easier and ensure timeliness of just using their own web pages.

The paper deals with the necessity of improving document creation processes in municipalities with support for Web applications. Given the great diversity of communities in terms of population, social and demographic composition is very difficult to analyze certain processes in the management of the village. Each village must have its members in number from 3 to 41 depending on the population as determined by law. It is evident that the larger towns and more representative bodies, the process of creating official documents of the village much more difficult process than in a small village and a small Members' Forum. In general, the smaller the municipality, the less is the official documents. If the issue of support of local government documents, viewed from the perspective of using web applications get very sad information. These findings warranted to improve our processes in communities emphasize and encourage us to further work in analysis and design solutions to this problem.

II. A BREAKDOWN PROCESSES IN THE ACTIVITIES, INPUTS, OUTPUTS AND TYPES OF DOCUMENTS

Municipalities manage them in the file form the following types of regulatory processes:

- Generally binding regulation
- Internal documents related to the operation of general office.

Model progress of activities related to the regulatory process in several steps:

- Preparation of draft document,
- To familiarize stakeholders with the prepared document,
- Observation,
- Inclusion of proposals,
- A proposal approver,
- Make comments,
- Incorporating comments,
- Proofreading,
- Approval,
- Registration,
- Publishing,
- Revision.

Inputs inducing changes in the regulatory process:

- Changes in legislation,
- Organizational changes,
- Change function,
- Rearrangement of functions.

The outcome of regulatory processes is always approved and published a public document, usually VZN, based on the Law on Municipalities No. SNR. 369/1990 Coll. Given that this law does not formally or content mandatory of those documents we outlined the process of creating these documents in the sequence diagram.

From the above sequence diagram shows that there is no use of IT or web technologies, apart from the fact that the documents prepared in word processor and then printed. Based on the diagram which are broken down by individual municipalities according to population visible move to a better, ie the number of municipalities with their own Web sites are increasing. Of the total 2780 villages with the population less 5000 has its own web page 2031 villages (Fig. 3). With a relatively large number of non-official sites. On the other hand, we have not found any municipality under 5000 inhabitants, where it was evident that the website used to support the process of community papers.

The diagram shows that the document will "electronic live " receipt of the draft mayor and then inserting it into the CMS. Document lifecycle is a continuous in a CMS, just depending on the ongoing process of either visible on the website or not. Through various user authorization can be elegantly generated document showing to all citizens or only to Members, or showing off at the time of treatment all groups.

Another big advantage is an effective reminder process, where you can visualize all the comments received, which can largely reduce duplication of designs and thanks to the availability through the web can be achieved by greater involvement of citizens of the village.

The process is to encourage compliance using CMS also has a major advantage and that is that the effectiveness of the designation may remain on the site until the eventual abolition, while over the Web is always available.

Since most CMS uses SQL database, we open up possibilities for further processing of official documents. Systematic addition of a selected document will gradually be created easily accessible database of internal documents of the village. Documents in paper form, this approach certainly does not print, but can radically reduce costs, the paper documents will be used only for archiving.

III Analysis of web sites of municipalities in the Slovak Republic

Slovak Republic consists of 2928 villages, which are grouped into 79 districts and 8 regions. The table shows the number of municipalities and districts in different regions. We assume that these figures provide information to analyze the possibility of using IT.

kód kraja	názov kraja	počet obcí	počet okresov
100	Bratislavský kraj	89	8
200	Trnavský kraj	251	7
300	Trenčiansky kraj	276	9
400	Nitriansky kraj	354	7
500	Žilinský kraj	315	11
600	Banskobystrický kraj	516	13
700	Prešovský kraj	666	13
800	Košický kraj	461	11

Figure 1 The distribution of regions, districts and municipalities

Of the 2,928 villages of 2,179 municipalities have own web site, of which 1781 is official. Of the total 749 municipalities have not own web site. By using community web site learn about their citizens living in the village and its problems, the direction of development and not least the legal documents relating to the citizens of the village. Unfortunately, this usually reduces the activity only to the disclosure of contracts and invoices, which the law prescribes.

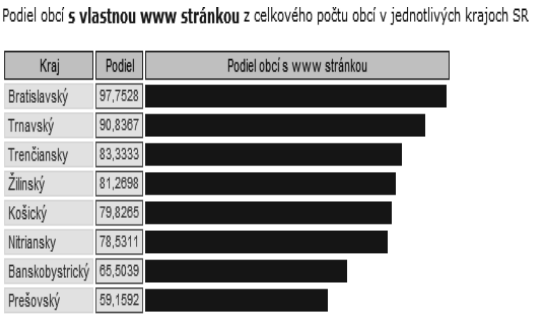


Figure 2 Share of municipalities with web site by region

The following chart shows the number of municipalities and the number of web sites in different districts. From the data provided it is clear that only a few districts have full coverage pages for each village and also can be seen that towards the east the situation is worse.

The following graph (In figure 3) shows the share of municipalities with web sites depending on the population. The analysis points to the fact that small municipalities have a problem with creating a web page. The larger the community, the easier own site generates. However, these communities share in the

total number of municipalities is significantly smaller. In addition to the main problem now is making the content of these pages.

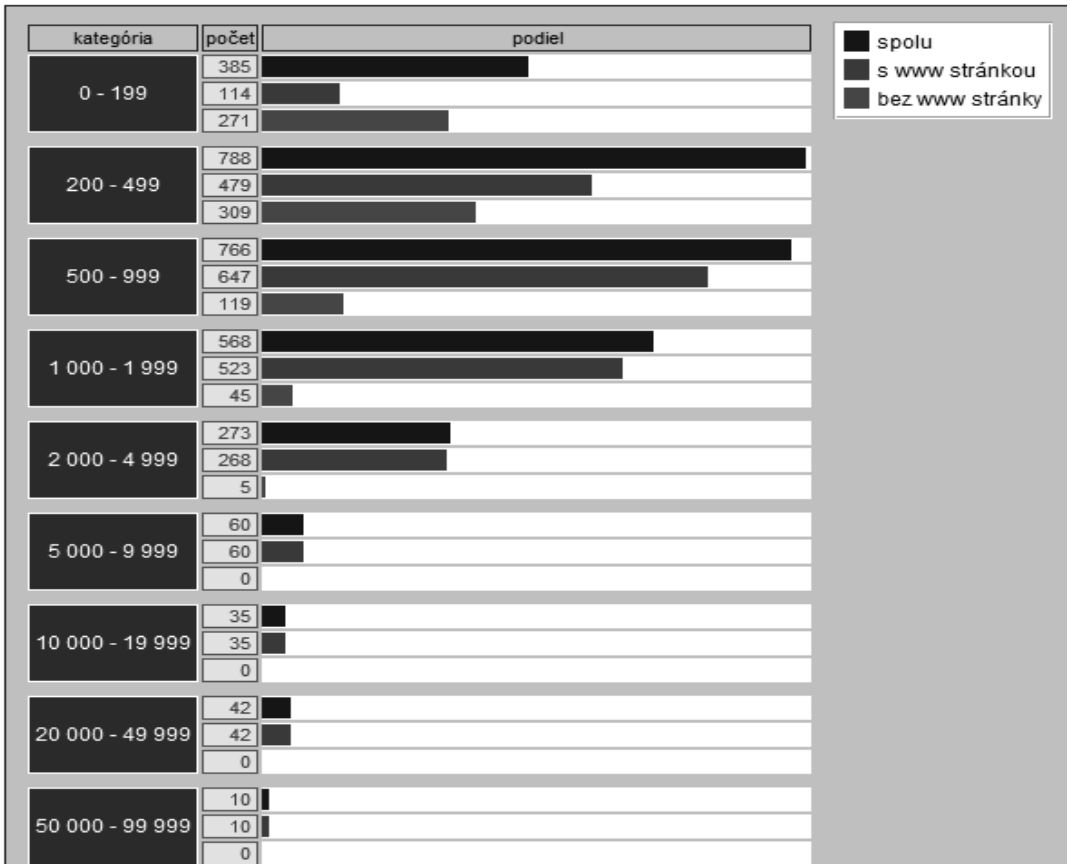


Figure 3 Share of municipalities with web pages depending on the population

IV. CONCLUSION

Based on the above data it is clear that in the near term, it is necessary to analyze the contents of the site of municipalities of the layout options, as well as the use of processes to support the village. The use of IT and web technologies specifically in the vast majority of municipalities unfortunately reduces only the disclosure of contracts and invoices under current legislation. Community legislation pushed the municipality to publish certain documents on the Internet, but still did not specify exactly how. Larger municipalities have purchased their own second-level domains and the smaller the order of tens or a few hundred people hired third-level domain. Over 700 municipalities without their own Web site to take care of other unspecified way. When hired third domain levels we can expect great support, web site content, by the provider. Rights, the small number of inhabitants in the village is the assumption that these communities may not have sufficient legal training citizen, working in the municipal office. To these villages would be a great asset creation IS, which would allow not only the production site but also the creation of documents and their publication in accordance with applicable legislation.

The situation in municipalities that have invested in buying your own domain and a webhost. These villages have the technical means, which generally do not because they do not know about them. Websites of these communities are mainly designed to elegantly modern CMS and regularly updated. Even for those municipalities would benefit from IS or software module that would allow documents to support the process of the village. This would of course re-engineering processes applied for the municipality, in accordance with applicable laws, therefore we

consider these processes re-engineering solution to be justified, indeed necessary.

REFERENCES

1. GRELL, M. - BANDURIČ, I. 2011. Nová koncepcia riadenia v územnej samospráve. In *Ekonomické aspekty v územnej samospráve : recenzovaný zborník príspevkov z vedeckej korešpondenčnej konferencie* [elektronický zdroj]. - Košice : Univerzita Pavla Jozefa Šafárika v Košiciach, 2011. ISBN 978-80-7097-863-4, s. 43-50. VEGA 1/0261/10.

2. Zákon SNR 369/1990 Zb. o obecnom zriadení, v znení neskorších predpisov

AUTHOR INDEX

Title Proceedings of the Eleventh International Conference on Informatics
INFORMATICS'2011

Date and Place November 16-18, 2011, Rožňava, Slovakia

Volume 1
Issue 1st issue

Editors Valerie Novitzká, Štefan Hudák

Year 2011

Imprint 90 pcs

Distribution
Department of Computers and Informatics
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 042 00 Košice, Slovakia
Phone: +421-55-63 353 13
Fax: +421-55-602 2746
URL http://kpi.fei.tuke.sk

ISBN 978-80-89284-94-8
EAN 9788089284948

Printing & layout adjustment
EQUILIBRIA, s.r.o., Poštová 13, 040 01, Košice, Slovak Republic

General sponsor

košice
iT VALLEY
association

Sponsors

FAKULTA HOSPODÁRSKEJ INFORMATIKY
EU v BRATISLAVE

ДГИНХ

ISBN 978-80-89284-94-8
9 788089 284948