

Intelligent Tutoring Interoperability for the New Web

M. Rosić, V. Glavinić*, S. Stankov

Faculty of Natural Sciences, Mathematics and Education, Split, Croatia

* Faculty of Electrical Engineering and Computing, Zagreb, Croatia

marko.rosic {stankov} .pmfst.hr

*vlado.glavinic@fer.hr

Abstract— Interoperability of systems based on knowledge is a very important element for reducing their development cost and enabling an easy-to-perform service enrichment. An important subclass of distant learning systems is represented by intelligent tutoring systems (ITSs) that base their work on the simulation of the "real" teacher in the learning and teaching process. ITSs base their interoperability on interchange of domain knowledge, knowledge about learning and teaching process and knowledge about students. This paper describes a methodology for ensuring interoperability among DiSNeT, a distance learning system based on the intelligent tutoring paradigm, on knowledge presentation using distributed semantic networks and on using agents in the learning and teaching process, with other ITSs.

I. INTRODUCTION

Development trends of present day information infrastructure based systems show a strong emphasis on ensuring interoperability, i.e. "the ability of two or more systems or components to exchange information and to use the information that has been exchanged" [1]. Such an approach enables both service enrichment as well as reduction of development and maintenance costs for the interoperating systems. Information exchange can be thus made possible by declaring knowledge in a common information space as well as by developing environments that are capable to resolve data requests. Agents from different systems approach this knowledge in order to perform their assignments.

It should be noted that ensuring interoperability implies not only information exchange but also information usage, what cannot be guaranteed without previously understanding information semantics. One of the features of the *Semantic Web* – the central constituent of the new Web generation – is enabling agents to understand semantics of data declared on the Web. Another feature of the new Web generation is enabling the design of systems having a service-oriented architecture, i.e. systems relying on Web services, which on the other hand enables assigning agents to systems. These agents jointly perform tasks so that every part of a system (agent) offers well-defined services to the other parts (agents). Consequences of both Semantic Web introduction and of service-oriented architecture imposition encompass semantically based browsing of Web published document contents, improved usage of Web based systems as well as better systems interoperability. In this way the typical process of

task solving augments into a value-adding chain where information is forwarded from one agent to another.

One in series of increasingly relevant classes of information infrastructure based systems is certainly that of distance learning systems, paving the way to e-learning service offering. These systems provide remote users a certain combination of services like teaching matter presentation, knowledge examination, recommendations for further work and collaborative learning, all upon a variety of interaction choices (e.g. student-student, student-teacher, student-educational institution).

The most important features of a distance learning systems environment include: distribution of data and processes execution, hypermedia orientation, dynamic system life cycle, open environment, diversity of processing platforms, variety of data presentation methods and storage and heterogeneity of users. An important subclass of distant learning systems is represented by intelligent tutoring systems (ITSs) that base their work on the simulation of the "real" teacher in the learning and teaching process.

As with every software system, the development of a distant learning system is an iterative process that can be represented by a suitable life cycle. On the other hand ITS development demands care to be used to specific elements that other distance learning systems do not have and is an expensive process securing the involvement of numerous experts from diverse fields like computer science, learning methodology as well as domain knowledge to be taught/learned. However, by enabling ITSs' interoperability reduction in development expenses can be achieved. ITS interoperability can certainly be enabled by networking their domain knowledge bases and/or sharing knowledge in a common information space.

II. NEW WEB GENERATION

A. *Semantic Web*

The Semantic Web is intended to impose structure to the meaningful content of Web pages, thus creating an environment where mobile agents roaming from page to page can readily carry out sophisticated tasks on behalf of their users. Hence it is but an extension of the current Web, in which information is given a well-defined meaning, better enabling computers and people to work in cooperation [2].

Defining ontologies on the Web enables introduction of content that is understandable to agents on the Web.

Ontology is the representation of certain domain categories and relations among categories. Defining ontology enables sharing "ways of understanding" among people and systems of a certain domain either being shared itself or providing domain knowledge for interchange. Considering the research previously being performed on ontologies ([3], [4]), it is obvious that their usage within systems based on knowledge can be represented by a three-layer model. In the lowest layer of this model languages are located that are used to define the ontology. Ontologies for different domains defined by languages from this layer are placed in the middle layer, while the topmost layer contains data described by the ontology from the middle layer. Information inside the same layer can be represented by different syntaxes. A definition of a certain ontology contains data like the following: (i) terminology of a domain being accessed by an agent, (ii) domain categories, (iii) relations among categories, (iv) category properties, and (v) inference rules. Both ontology definition and data described by it are enclosed in a Web document's content. An agent accessing this data has to know some ontology definition language (belonging to the lowest layer in the ontology model). Before accessing data, the agent accesses the ontology definition, acquires the ontology itself and eventually accesses the data been declared using this ontology. This way of accessing data is impossible without complying with standards that define both ontology definition and data representation. Presently there are several models that support the above elements, here including ([5], [6]): Ontology Inference Language (OIL), DARPA Agent Markup Language (DAML), Simple HTML Ontology Extension (SHOE), Resource Description Framework (RDF) + Resource Description Framework Schema (RDFS), Web Ontology Language (OWL), and Ontobroker.

Fig. 1 shows a part of an ontology represented using the SHOE model. It shows the ontology that defines categories *person*, *student* and *teacher*. The category *person* is starting category. The categories *student* and *teacher* are related to the category *person* with a relationship of type *IS_A* thus indicating that they have all the properties belonging to the category *person* (properties *name* and *surname*). Furthermore, the ontology in Fig. 1 has defined relation *advisor* that assigns *teacher* to *students*. By accessing the ontology defined in Fig. 1, an agent gets the information about the domain where it has to fulfill its tasks. Only after getting this information, the agent can access data declared by the ontology. Acquiring the ontology from Fig. 1 enables using data shown in Fig. 2. Identifiers are assigned to described objects. The example in Fig. 2 describes the object *student* that has the identifier *www.pmfst.hr/#IAntic*. The Semantic Web uses the Universal Resource Identifier (URI) mechanism for assigning identifiers to objects that are described.

Data defined in Fig. 2 belongs to the topmost layer of the three-layer ontology model, the ontology from Fig. 1 belongs to the middle layer while the SHOE model belongs to the lowest layer of this model, i.e. to the ontology definition languages layer.

Gruber [7] divides ontologies as follows: (i) domain ontologies, (ii) task ontologies, (iii) top-level ontologies, and (iv) system ontologies.

```
<HTML>
...
<BODY>
<ONTOLOGY ID="users_ontology"
VERSION="1.0">
<USE-ONTOLOGY ID="base-ontology"
VERSION="1.0" PREFIX="base"
URL="http://www.cs.umd.edu/projects/plus/
SHOE/base.html">
  <DEF-CATEGORY NAME="Person"
ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="Student"
ISA="Person">
<DEF-CATEGORY NAME="Teacher"
ISA="Person">
  <DEF-RELATION NAME="Name" >
    <DEF-ARG POS="1" TYPE="Person">
    <DEF-ARG POS="2" TYPE=".STRING">
  </DEF-RELATION>
<DEF-RELATION NAME="Surname" >
    <DEF-ARG POS="1" TYPE="Person">
    <DEF-ARG POS="2" TYPE=".STRING">
  </DEF-RELATION>
<DEF-RELATION NAME="Advisor" >
    <DEF-ARG POS="1" TYPE="Student">
    <DEF-ARG POS="2" TYPE="Teacher">
  </DEF-RELATION>
</BODY>
</HTML>
```

Figure 1. Part of an ontology

```
<HTML>
...
<BODY>
<USE-ONTOLOGY ID="users_ontology"
URL="http://www.pmfst.hr/~marko/users_ont
ology.html" VERSION="1.0" PREFIX="uo" />
<CATEGORY NAME="uo.Student"
FOR="www.pmfst.hr/#IAntic" />
<RELATION NAME="uo.Name">
  <ARG POS="1" VALUE="TO"/>
  <ARG POS="2" VALUE="Ivan"/>
</RELATION>
<RELATION NAME="uo.Surname">
  <ARG POS="1" VALUE="TO"/>
  <ARG POS="2" VALUE="Antic"/>
</RELATION>
<RELATION NAME="uo.Advisor">
  <ARG POS="1" VALUE="TO"/>
  <ARG POS="2" VALUE="
www.pmfst.hr/#Alvanic"/>
</RELATION>
</BODY>
</HTML>
```

Figure 2. Data described by ontology

Domain ontologies contain categories related to certain domains and describe domain terminology, domain elements, relationships among elements, properties of elements and some additional information related to a certain domain. *Task ontologies* contain categories related to problem solving (e.g. medical diagnostics). Domain ontologies and task ontologies can share common *top-level ontologies* containing categories related to both of them. Categories that top-level ontologies relate to are those of space type, time, units of measure etc. *System*

ontologies are the most complex ones because they contain domain ontology elements as well as task ontology ones, which needed for the realization of functions of the system using that ontology.

B. Web Services

The issue that is often to be solved when designing modern distributed systems is the access to distributed data. This is usually accomplished by employing one of the existing *middleware* based technologies. Middleware is the intermediate software in charge of integration of distributed system objects, i.e. it has to insure a proper and meaningful interaction of distributed system objects [8]. In the specific case of distributed object based middleware, this intermediary is often realized by a kind of *Object Request Broker* (ORB), whose main functions of encompass (i) locating and activating the called objects and (ii) enabling the communication among them. Examples of architectures incorporating the ORB concept are Common Object Request Broker Architecture – CORBA and Distributed Component Object Model – DCOM. Similar principles underlie the implementation of distributed systems using Service-Oriented Architecture – SOA, where instead of being objects based the systems are agents based. Agents in these systems jointly execute tasks so that every part of a system (i.e. an agent) provides well-defined services to the other parts (i.e. agents). The principal elements of the service-oriented architecture [9] are (i) agents that enable or demand services, (ii) messages that interchange by them, (iii) mechanisms for message transmission, as well as (iv) mechanisms for service description. Agents communicate through message interchange, while the service-oriented architecture has to insure the mechanisms for messages transmission as well as those for service description. A service in this architecture is considered to be an agent that both performs well-defined operations and could be asked to execute its tasks using some defined mechanisms. Developers building systems based on such services consider them through both their interfaces and their intercommunication using the respective interfaces. Service-oriented architecture implementations within the Web environment are Web Services. According to W3C (the WWW consortium), a Web Service is the software designed to support common work of networked systems by enabling their interaction. Specific entities that enable these services are agents. Web Services can be described as abstract sets of functions, while their realization can be described as agent systems [9]. The main elements enabling Web Services are [10]: (i) the Web Services Description Language (WSDL), (ii) the Simple Object Access Protocol (SOAP), and (iii) the Universal Description, Discovery and Integration (UDDI). WSDL is the language for describing Web Services, which is based on XML descriptions. SOAP is the protocol for inter-agent message transmission, which is also XML based. UDDI is the method for publishing Web Services descriptions within the Web information space that enables finding the appropriate service provider. The definition of Web Services does not specify how are to be implemented agents offering Web Services. It only

defines the way messages transmitted among service applicants and their providers, and the way services are described and declared.

III. DISTRIBUTED SEMANTIC NETWORK TUTOR (DiSNET)

Distributed Semantic Network Tutor (DiSNeT) is a distance learning system based on the intelligent tutoring paradigm, on knowledge presentation using distributed semantic networks and on using agents in the learning and teaching process. Intended system users are experts, teachers and students. Experts create expert domain knowledge, while teachers create courseware to be used in the learning and teaching process by students. The architecture of DiSNet encompasses the next four modules, see Fig. 3: (i) Knowledge module, (ii) Authoring module, (iii) Teaching module, (iv) Support collaboration module. The *knowledge module* is the fundamental module of DiSNeT and contains the knowledge used by all other modules of the system. The module comprises expert knowledge bases along with educational knowledge bases that are created using the authoring module or gained using the support collaboration module. Knowledge stored in the knowledge module uses the teaching module that enables the learning and teaching process by publishing both stored knowledge and student profile data. The *authoring module* supports the creation of both expert knowledge elements and educational knowledge elements. Expert knowledge contains domain knowledge, which is represented by distributed semantic networks. Here every node can contain the following elements: (i) expert knowledge node identifier, (ii) expert knowledge node name, (iii) table containing node properties and respective values, (iv) node description, (v) paths for files describing node hypermedia, (vi) list of expert knowledge node identifiers, which are linked to a certain node as well as the names of the respective links. DiSNeT can use expert knowledge created using the intelligent hypermedia authoring shell Tutor Expert System (TEx-Sys) [11]. The authoring module also incorporates the Web Service used by author agents to add or change the knowledge. The *teaching module* is implemented using the Web Service used by teaching agents assigned to students. This Web Service enables the access to the knowledge being taught and to the knowledge about students. Using the published knowledge, teaching agents guide the learning and teaching process of a distant student. The implementation of teaching agents and the details of their teaching paradigm are left to agent developers, what enables the design of different ITSs on the services of a single teaching module. Developing different ITSs on teaching and authoring module's Web Services ensures ITS interoperability. By doing this, the DiSNeT server becomes the information space for interchanging knowledge belonging to different ITSs, which is the basis for their tutoring activity. Another way to achieve ITS interoperability is the use of the Web as a whole as the common information space. This kind of interoperability is enabled by the *support collaboration module*, whose functions can be subdivided into two groups. The first group comprises functions for declaring expert and educational knowledge while the second comprises functions for searching and unifying knowledge in the

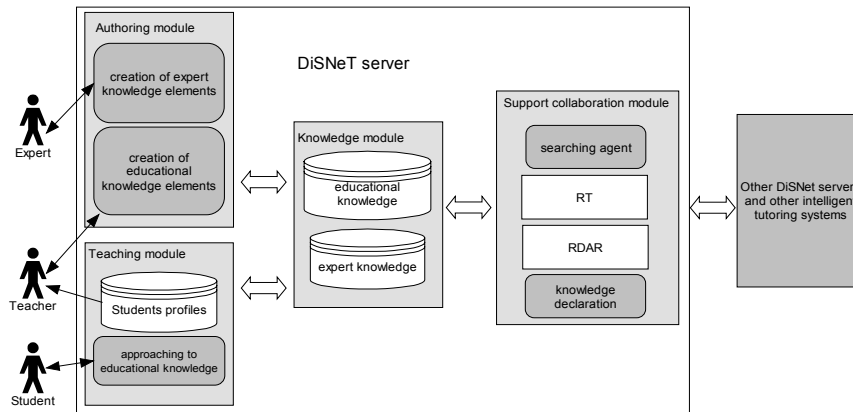


Figure 3. DiSNeT's structure

common information space. Educational and expert knowledge are published on the Web using ontologies previously developed according to the SHOE model. Publishing expert and educational knowledge using appropriate ontologies enables knowledge interchange not only with distance learning systems based on DiSNeT, but also with other distance learning systems that adopt the ontologies so far defined. The search part of the support collaboration module contains the search agent, the requests table (RT) and root documents address register (RDAR) pointing to expert and educational knowledge. The root documents address register contains addresses of those documents that represents the root of the tree of networked documents, while addresses of other documents are reached after analyzing the content of previously browsed documents, which stored in the requests table. The content of the root documents address register is regularly checked to discover documents that are no longer available thus enabling the removal of the respective addresses from the register. If the search agent finds a new document in the root documents address register, it requests its distribution, analyzes its content and stores within the knowledge module the knowledge elements thus found. A new document can be connected to other documents and the search agent analyzes their content as well, adding new data to the expert and educational knowledge. Multimedia elements that describe teaching matter are not stored on the DiSNeT server but only their addresses. The search agent completes its task after analyzed both all new root documents and all documents from the requests table. Additionally, search agent also analyzes the syntax correctness of the published documents.

IV. CONCLUSION

This paper describes a methodology for ensuring interoperability among DiSNeT, a distance learning system based on the intelligent tutoring paradigm, on knowledge presentation using distributed semantic networks and on using agents in the learning and teaching process, with other ITSs. ITS interoperability within such a framework lies upon Web Services and Semantic Web artifacts. By using Web Services DiSNeT allows both creating new knowledge and interchange of knowledge stored in its knowledge module. The system uses the Web as the information space for interchanging knowledge with other ITSs. This is enabled by the support

collaboration module, which performs a bi-directional interchange of knowledge: (i) declaration of DiSNeT system knowledge using common ontologies, and (ii) searching the Web for knowledge published by other ITSs.

ACKNOWLEDGMENT

The work described in this paper has been carried out within projects 0036033 *Semantic Web as Information Infrastructure Enabler*, and 0177110 *Computational and Didactical Aspects of Intelligent Authoring Tools in Education*, both funded by the Ministry for Science and Technology of the Republic of Croatia.

REFERENCES

- [1] xxx: *A compilation of IEEE standard computer glossaries*, Institute of Electrical and Electronics Engineers, New York, 1990.
- [2] T. B. Lee, J. Hendler, O. Lassila: "The Semantic Web", *Scientific American*, May, 2001.
- [3] R., Jasper, M. Uschold: "A Framework for Understanding and Classifying Ontology Applications", *Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods*, Stockholm, Sweden, August, 1999.
- [4] D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, P. Patel-Schneider: "OIL: An Ontology Infrastructure for the Semantic Web", *Intelligent Systems*, Vol. 16, No. 2, 2001.
- [5] J. Heflin, J. Hendler, S. Luke: "SHOE: A Knowledge Representation Language for Internet Applications", *Technical Report, CS-TR-4078 (UMIACS TR-99-71)*, Dept. of Computer Science, University of Maryland, USA, 1999.
- [6] S. Laks, F. Sadri: "XML Interoperability", *Proceedings of the Workshop on Web and Databases (WebDB)*, San Diego, California, USA, June, 2003.
- [7] T. Gruber: "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", *International Journal of Human and Computer Studies*, Vol. 43, No. 5-6, pp. 907-928, 1995.
- [8] W. Emmerich: "Software Engineering and Middleware: A Roadmap", in A. Finkelstein (Ed.): *The Future of Software Engineering*, ACM Press, 2000.
- [9] xxx: "Web Services Architecture: W3C Working Draft 8", 2003., <http://www.w3.org/TR/2003/WD-ws-arch-20030808/>
- [10] J. Korhonen, L. Pajunen, J. Puustjarvi: "Using Web services and workflow ontology in multi-agent systems", *Proceedings of Workshop on Ontologies for Multi-Agent Systems*, Sigüenza, Spain, September 29-30, 2002.
- [11] S. Stankov: *Isomorphic Model of the System as the Basis of Teaching Control Principles in an Intelligent Tutoring System*, PhD Diss., Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Croatia, 1997., (in Croatian)