# Semantic RIA (Rich Internet Application) Configuration Model

**Zoran Balkić**
Faculty of Electrical Engineering
J.J. Strossmayer
University of Osijek
Osijek, Croatia
zoran.balkic@etfos.hr

**Ivica Lukić**
Faculty of Electrical Engineering
J.J. Strossmayer
University of Osijek
Osijek, Croatia
ivica.lukic@etfos.hr

**Ivan Aleksi**
Faculty of Electrical Engineering
J.J. Strossmayer
University of Osijek
Osijek, Croatia
ivan.aleksi@etfos.hr

*Abstract* — **The major goal of this paper is to introduce semantic approach for building RIA focusing on the web application semantic configuration model. Due to very large number of available ontologies for different knowledge domains, no suitable models were found for web application configuration, which could be reused, therefore opening the space for defining one, using the minimalistic approach and enabling web application integration at configuration level.**

*Keywords* — **Semantic Rich Internet Application (RIA), semantic configuration model**

## I. INTRODUCTION

Since the initial ideas of the semantic web approach different ontology modeled solutions were built. By the nature they cover very large domain knowledge from the perspective of human activity, medical issues, biological, technical and geospatial views, etc. What is new today is normally taken as granted and easily accepted tomorrow by users. Simply to say, users always want more. Today they want up to date information and they want to access it according to their point of view or particular preferences. To cope with these demands, websites must be dynamic and must be able to reconfigure automatically their structure, content and appearance. This scenery has favored the creation of tools for automatic generation and management websites. A lot of, today standardized vocabulary components from Dublin Core, Creative Commons, FOAF, GeoRSS, MediaRSS, and others are already supported by Yahoo semantic search engine thus allowing smart, fine grained, reusing project domain knowledge. Many services will support RDFa and eRDF markup to embed semantics into existing HTML pages as well. This paper deals with RIA (Rich Internet Application) presentation and configuration mechanism for seamless semantic project modeling. It was created with one thought in mind, to be an upper configuration ontology and inspiration for further development.

## II. MODELING BACKGROUND

### A. RIA - Configuration coverage

Taking into consideration the state of the art Web GUI (Graphical User Interfaces) that are in mass use today ontology concepts seems to be the perfect choice for RIA configuration challenge. In the area of Information Systems, ontology is defined as a set of concepts and terms, that can be used to describe some area of knowledge, or construct a representation of the knowledge [ST99]. According to Chandrasekaran [Cha99], ontologies are theories of content about the object types, object properties, and relationships between objects that are possible in a domain of specific knowledge. When we refer to the semantic web we are thinking about a network of concepts, respectively. Each concept has a group of related resources and can be related to other resources, thought we can use this concept network (also called ontology) to navigate among web resources or simply among information resources. The main idea behind is to integrate the specification of these concept network ontologies with the navigation and their storage. We intend to do this using XML (RDF/OWL) technology in every component of the system.

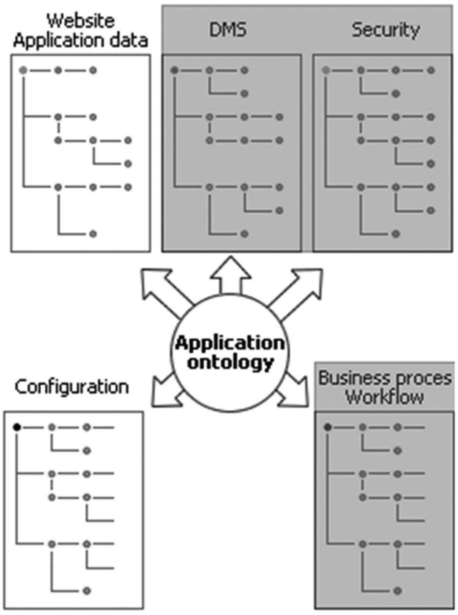


**Figure 1.** Application framework configuration ontology definitions [1]

The basic building brick is the core of the system for storage and retrieval of data and metadata. In our scenario it is JCR (Java Content Repository) as data manipulation layer, basically built on top of the tree representation for the data structures that corresponds with semantic way of analysis and modeling emphasizing the use of URIs for data creation/retrieval [1] based on the REST principles as explained in [3].

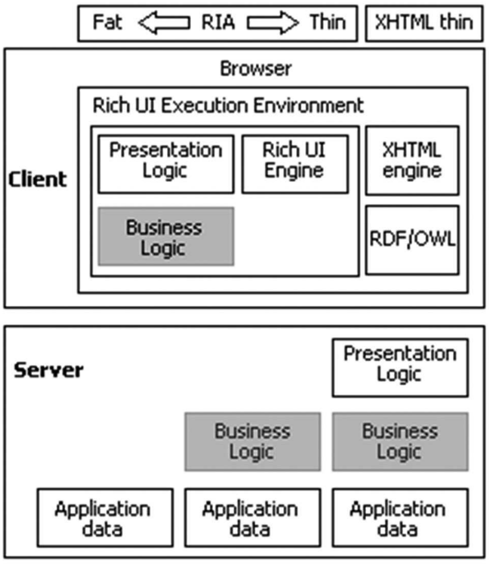


**Figure 2.** RIA Client-Server layering

### *B. RIA – Configuration Classes and instances*

In the description of particular ontologies, authors often rely heavily on a presentation of the basic taxonomy of entities in the world to focus the discussion. The formation of this taxonomy may not be the most important dimension of ontology design; however, it provides a backbone from which domain specialists can then extend upper ontologies. For this reason we will first go into some detail concerning the formation of taxonomies and their practical correlation. Mathematically, taxonomies are lattice structures containing ontological classes as nodes. We shall use the term 'class' to refer to those entities in an ontology that represent the concepts or categories found in the specific domain that is being modeled.

Modeling domain should cover the necessary RIA building blocks that consists of domain concept divided into four parts:

1. Presentation model which includes:

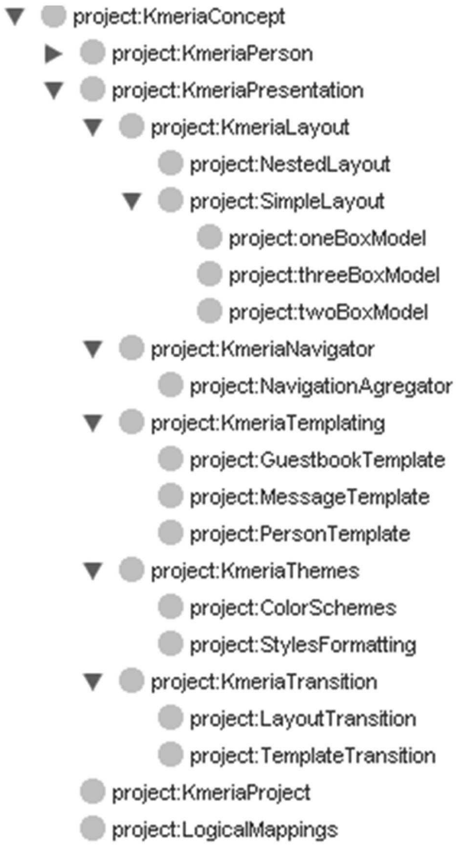
- Layout models (models for usage in RIA and XHTML rendering pipeline) – specific layout models for dynamic GUI presentation generation which is consisted out of nested and simple layout elements. Simple layout elements define basic presentation models
- Navigator model (includes custom Application data views – smart navigation which is part of the published REST interface for default views)
- Templating model (includes definition of the Application data model view). Each Master / Detail view with multiple (synchronized) views for the Application Components (in our case Guestbook Application Model Classes)
- Themes model (covers the Color and Styling scheme for RIA) applicable to any Templating model or project specific nodes
- Transition model (binds on stage layout and template transitions)

2. Project description

- Covers project specific configuration which is binded to the Application data model for automatic rendering process

3. Logical mappings (used by the system and other custom defined Logical mappings – or nested logical Navigator model)

4. Application user (Application configuration Person, both registered and not registered)



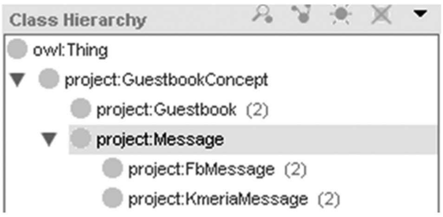
- ▼ ● project:KmeriaConcept
  - ▶ ● project:KmeriaPerson
  - ▼ ● project:KmeriaPresentation
    - ▼ ● project:KmeriaLayout
      - ● project:NestedLayout
      - ▼ ● project:SimpleLayout
        - ● project:oneBoxModel
        - ● project:threeBoxModel
        - ● project:twoBoxModel
    - ▼ ● project:KmeriaNavigator
      - ● project:NavigationAgregator
    - ▼ ● project:KmeriaTemplating
      - ● project:GuestbookTemplate
      - ● project:MessageTemplate
      - ● project:PersonTemplate
    - ▼ ● project:KmeriaThemes
      - ● project:ColorSchemes
      - ● project:StylesFormatting
    - ▼ ● project:KmeriaTransition
      - ● project:LayoutTransition
      - ● project:TemplateTransition
  - ● project:KmeriaProject
  - ● project:LogicalMappings

**Figure 3.** RIA Project Configuration asserted hierarchy

For ontology sub domains that are Application and Configuration concept two distinct basic configuration parameters are defined, at this moment taking into consideration only framework default configurations (project Configuration).

### *C. RIA – Application sub domain Classes and instances*

For the purpose of presentation we have chosen the simple Guestbook Application domain model which consists of well known model for anonymous user Guestbook creation and retrieval. In this particular case domain definition is consisted of two classes (Guestbook and Messages – that can be either internal or external in nature).



Class Hierarchy

- owl:Thing
- project:GuestbookConcept
  - project:Guestbook (2)
  - project:Message
    - project:FbMessage (2)
    - project:KmeriaMessage (2)

**Figure 4.** RIA Application sub domain class hierarchy

Application domain supports adding guestbook as project specific nodes which consists either of native guestbook messages entries or external messages format (such as facebook message for example). Each message has it own set of properties and domain classes that corresponds with default template rendering mechanism.

## III. PRACTICAL SOLUTIONS

As a practical approved solution in the software modeling and engineering tools standard MVC pattern for application is used, as shortly described in [4]. Where standard MVC architecture is applied for classic Web application models, in practical usage several issues arise with usage of pure MVC pattern Application approach. Issues with pure MVC:

- MVC only applied to special GUI components (e.g. tables)
- Model represents business state - Presentation state not modeled
- Models and ACs get mixed up - E.g.: AC Messages List must reference AC Guestbook Catalog in order to access the selected messages
- Result: ACs are not decoupled - E.g.: AC Message List cannot be developed / tested / used without AC Guestbook catalog

From the simplicity perspective GUI components are mapped with respective Application component templating (Guestbook, Message, and Person) which is mutually compatible which allows us to reuse template parameters outside of the appropriate sub domain (message -> message templates). This approach actually applies application MVC pattern to each one of Application component models.

In the proposed example an evaluation of the simple upper ontology for defining RIA configuration is presented allowing RIA applications interoperability at application configuration level. Further ontology mapping can be done with external upper ontologies developed for the specific Application domain.

## IV. CONCLUSION

On the present Web application frameworks, the lack of semantics prevents computer systems from being able to interpret Web information automatically. For a task such as Web-based RIA generation and delivery we have introduced the overall configuration model blocks for building such an application.

Deployment features are built in application ontology, thus, providing us with the flexibility and code development. Integration techniques are built in the system as a native part from the modeling stage to the exploitation. Modular architecture provides separated computational artifacts (Application data and configuration model) that can be, in the future, deployed as standard generic Web services models which can be used with other semantic approaches to this problem.

The hypothesis that a Semantic Web based modeling might be particularly suitable for knowledge aggregation in a dynamic RIA environment is evaluated with a simple prototype migration implementation. All modeling ontologies and Full Paper can be found and downloaded at http://www.kmeria.com/ontologies/.

## V. ACKNOWLEDGMENT

I would like to thank to my dear friends and colleagues, Akos Papp and Sasa Vidakovic employed at Green Studio Ltd. that spent a lot of time with me, contemplating, programming, searching and finding the easiest and simplest solution for the Semantic Rich Internet Application modeling and development.

## VI. References

[1] Zoran Balkić, Marina Pešut, Franjo Jović, Semantic Rich Internet Application (RIA) Modeling, Deployment and Integration

[2] John Beatty, BEA Systems, Stephen Brodsky, IBM Corp.,Martin Nally, IBM Corp. Rahul Patel, BEA Systems, Next-Generation Data Programming: Service Data Objects, A Joint Whitepaper with IBM and BEA

[3] Roy Thomas Fielding, Architectural Styles and the Design of Network-based Software Architectures (REST)

[4] MVC architectural pattern, http://en.wikipedia.org/wiki/Model-view-controller

[5] Digital Enterprise Research Institute (DERI), http://www.deri.org

[6] OMG: Ontology Definition Metamodel (ODM), http://www.omg.org/cgi-bin/doc?ad/06-05-01.pdf.

[7] JCR Implementations, Communique – http://www.day.com, Apache Jackrabbit – http://jackrabbit.apache.org

[8] Protégé approach and toolset http://protege.stanford.edu/

[9] OWL delivery language http://www.w3.org/2004/OWL

[10] Guha, R. 2002. Tap: Towards the semantic web. Demo on World Wide Web 2002 Conference. At: http://tap.stanford.edu/www2002.ppt.

[11] Hayes, P. 2003. RDF semantics. Proposed Recommendation. http://www.w3.org/TR/2003/PR-rdf-schema- 20031215/.

[12] Hyv¨onen, E., Saarela, S., Viljanen, K.: Application of ontology-based techniques to view-based semantic search and browsing.

[13] Roger Harris, Peter Wiebe, Jurgen Lenz, ICES Zooplankton Methodology manual, Acadamic Press 2000.