

Sabin Buraga¹, Marius Cioca²

¹Faculty of Computer Science, “A.I. Cuza” University of Iași, Romania
busaco@infoiasi.ro

²Faculty of Engineering, “L. Blaga” University of Sibiu, Romania
marius.cioca@ulbsibiu.ro

Abstract

The paper presents the opportunities given by the Semantic Web technologies for the enterprise integration in the context of development of e-business software applications.

Introduction

In any enterprise, there are certain “IT-oriented islands” – legacy applications, database management systems (e.g., database relational management systems), client programs, etc. – hard to be incorporated in a consistent semantic-based manner. The integration of enterprise with its suppliers and customers or with entities from different geographical regions is more difficult, because the transferred information (documents, multimedia streams, user’s data, etc.) is very complex. Due of the complexities above mentioned, for a good integration of the business activities, we must find a coherent manner of modeling the main concepts of the e-enterprise and the relationships that can be established between the components of the enterprise (Cioca, 2004).

The paper will investigate the possibility to use different standardized semantic Web-based languages, such as RDF – *Resource Description Framework* (Beckett, 2004; Buraga, 2004a), DCMI – *Dublin Core Metadata Initiative* (Buraga, 2001), and OWL – *Web Ontology Language* (Buraga, 2004a; Smith *et al.*, 2004), in order to model a complex software system to be deployed within an e-enterprise and to enhance the inter-connectivity between the heterogeneous components of that system. Starting with an UML – *Unified Modeling Language* (Booch, Rumbaugh & Jacobson, 1999) specification, we can automatically generate XML-based representations of the e-enterprise’s main components (Cioca & Buraga, 2003; Buraga & Cioca, 2004).

Levels of enterprise integration

In order to integrate various components of an enterprise, we need to address the following levels of integration (see also figure 1): *system integration*, *application integration*, *business integration*, and *enterprise integration*.

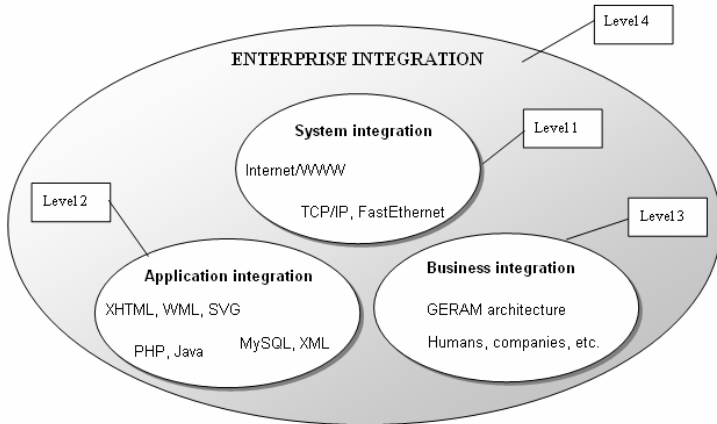


Figure 1. Levels of enterprise integration

System integration

The system integration deals with the low-level communication components, such as network services and protocols.

Application integration

A traditional enterprise-oriented Web application can consist of some distributed Web components, such as CGI (*Common Gateway Interface*) scripts written in C, bash, Perl or Python languages, Web application servers (notably, PHP, JSP or ASP.NET), Web services and/or Web agents. From the point of view of data sources, the designed system implies the existence of multiple database relational servers (e.g., Oracle, MySQL or PostgreSQL servers), workflow and dataflow monitors, XML-based documents (on the server, the content can be easily stored into XML documents without the layout or even within *native XML databases*, such as Berkeley XML DB, eXists or Apache Xindice), and plain text files (e.g., used for storage of configuration parameters or log information).

The Web interface needs to comply with a standard for inter-operability, due of multiple browsers: classical desktop graphical-oriented and HTML-based browsers (such as Mozilla Firefox or Opera), text-oriented browsers (e.g. Lynx) or WML (*Wireless Markup Language*) browsers that can be used in a wireless environment.

More over, in order to present the graphical information, some resources can be written in several XML-based languages. For example, to create vector-based sales charts we can use SVG – *Scalable Vector Graphics* (Buraga, 2004b; Jackson, 2004), a standardized recommendation of the World-Wide Web Consortium (W3C, 2005).

Business and enterprise integration

In order to assure business integration, we choose GERAM architecture (Cioca, 2004). One of the main advantages of the GERAM methodology is the great level of detail that can be obtained in order to describe all elements of the enterprise. Another reason why it had been opted for the GERAM architecture is its adaptability.

At this level, we need a modeling language that can be used to design the whole conceptual structure of an enterprise and its inter-connections to other related enterprises. We opted for the powerful and largely accepted UML language (Booch, Rumbaugh & Jacobson, 1999). UML is a language for specifying, visualization, building and documenting the software systems products, and also for enterprise modeling or other non-IT systems.

Using Semantic Web technologies to describe the e-enterprise

The actual WWW space is primarily composed on pages (markup documents) with information in the form of natural language text and multimedia intended for humans to read and to understand. Computers are used principally to render this hypermedia information, not to reason about it. Information retrieval has become omnipresent and information needs no longer to be intended for human readers only, but also for machine processing, enabling intelligent information services, personalized Web sites, and semantically empowered search engines – this is the seminal idea of *Semantic Web* (Davies *et al.*, 2003).

Semantic Web technologies are based on the XML (*Extensible Markup Language*) and is structured on three main layers (see figure 2):

- *the metadata layer* offers an extensible framework in order to express simple semantic assertions (e.g., vocabularies or taxonomies); this conceptual model can be used to attach metadata (data about data) to each Web resource;
- *the schema layer* can help to specify simple ontologies in order to define a hierarchical description of the concepts and properties for a given resource;
- *the logical layer* introduces ontological languages that are capable to model complex ontologies; at this layer, in the future will exist different *reasoning services* to be used by the applications oriented to Semantic Web.

Using metadata

Each component of the e-enterprise system can be described with the help of metadata. The *metadata level* is the first level of a semantic Web-based application (Buraga, 2004). This metadata can be attached to each software component in order to store several important characteristics (e.g., information regarding uptime, ownership, execution platform, etc.).

Also, for each user we can retain the information about his/her status (e.g., the user role – administrator, database manager, security monitor, regular user, visitor etc. – within the system), personal data (for example, age, user e-mail address, location, etc.), user-interface preferences (layout, chromatic and interaction preferences, etc.). To associate and store metadata, we use RDF (Beckett, 2004; Buraga, 2004a) – a model for processing metadata. RDF standard provides interoperability between applications that exchange machine-understandable information on the World-Wide Web. RDF is intended to be used to capture and express the conceptual structure of information offered by the Web.

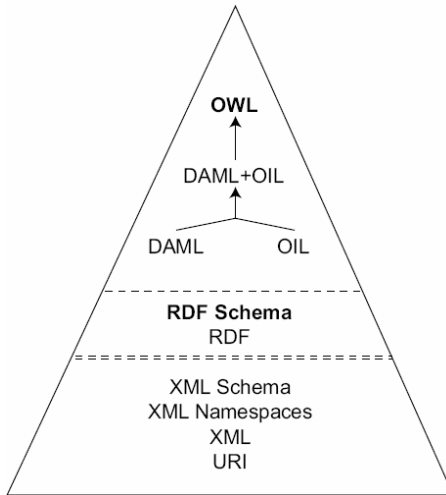


Figure 2. The pyramid of Semantic Web languages (Geroimenko, 2004)

Expressing relations between e-enterprise's components

The relations established between the components of the system can be also expressed by RDF assertions. For example, we can trace a relation of dependence between different e-enterprise modules (e.g., user authentication module, workflow monitor, etc.) and the involved resources. The relations are useful to maintain the connection between the e-enterprise application and its users, in order to develop a proper human resources management software tool (Cioca & Buraga, 2003). Using FOAF – *Friend Of A Friend* (FOAF, 2005) statements, we can create the graph of inter-connections between the persons that participate into a task or use a certain component (such as the project management software tool).

Using ontologies

A superior level of modeling is to create or (re)use *ontologies* to represent the knowledge within the application. The semantic structure achieved by ontologies differs from the superficial composition and formatting of information (viewed as data) afforded by relational and native XML databases. Ontologies are able to provide an objective specification of domain information (in our case, the e-enterprise domain) by representing a consensual agreement of the concepts, characteristics, and relations characterizing the way knowledge in that domain is expressed (Daum & Merten, 2003; Si Shi & Murthy, 2003).

The RDF – *Resource Description Framework* (Beckett, 2004) and OWL – *Web Ontology Language* (Smith et al., 2004) facilities are decisive in modeling of a knowledge-based e-enterprise system. There are several upper-level ontologies – such as the *ABC ontology* (ABC, 2001) – can be used to model, at the enterprise level, the concepts, properties and relations between these concepts. The ABC ontology has been designed as a primitive (general) ontology so that individual communities are able to build on top of it more specific ontological constructs. The primitive category at the core of the ABC ontology is an entity. At the next level three main categories are offered: *temporality*, *actuality* and *abstraction*.

Assuring the inter-connectivity of the e-enterprise's components

Also, we can use the ontological statements expressed by RDF and OWL languages to enhance the inter-connectivity between the components of the given application.

One of the major concerns of the e-enterprise application's analysts, designers and programmers is to use a platform- and language-independent protocol for exchanging information between the software components (scripts, agents, Web services, etc.) of the e-enterprise system or between these components and other entities, i.e. final users (clients), suppliers, processes, modules. The interactions between the users or between the users and the software components comply with a standard for interoperability (Buraga & Alboaie, 2004). One solution is to consider an XML-based protocol, such as SOAP – *Simple Object Access Protocol* (Mitra, 2003). SOAP is a simple lightweight protocol that can be used for structured and strong-type information exchange in a decentralized, distributed environment. The SOAP messages can encapsulate “intelligent” data expressed by XML constructs (metadata, schemas, ontologies, etc.).

Using the process ontology described by the OWL-S specification (OWLS, 2004), we can model the relations between the components of the e-enterprise application and to integrate these components based on their characteristics (classes) on the semantic basis.

Conclusion

The paper presented the solution of modeling certain characteristics of an enterprise by using Semantic Web languages. Following this proposal, a more conceptual and straightforward approach can be applied to model and build complex e-business software applications.

References

- D. Beckett (ed.), *RDF/XML Syntax Specification (Revised)*, W3C Recommendation, Boston, 2004: <http://www.w3.org/TR/rdf-syntax-grammar>
- G. Booch, J. Rumbaugh, I. Jacobson, *UML User Guide*, Addison-Wesley, 1999
- S. Buraga, *Semantic Web* (in Romanian), Matrix Rom, București, 2004
- S. Buraga (coord.), *Situri Web la cheie* (in Romanian), Polirom, Iași, 2004
- S. Buraga, *Tehnologii Web* (in Romanian), Matrix Rom, București, 2001
- S. Buraga, L. Alboaie, S. Alboaie, *An XML-based Serialization of Information Exchanged by Software Agents*, International Informatica Journal, 28 (1), 2004
- S. Buraga, M. Cioca, *Using XML Technologies for Information Integration within an E-Enterprise*, Proceedings of the 7th International Conference on Development and Application Systems, Suceava University Press, 2004
- M. Cioca, *Conducerea asistată a unităților economice* (in Romanian), Editura Universității “L. Blaga”, Sibiu, 2004
- M. Cioca, S. Buraga, *New Tools for Human Resource Management in e-Business: Combining UML Language, Reference Architectures and Web Programming*, Proceedings of International Conference on Industrial Informatics, R.Unland *et al.* (eds.), IEEE Industrial Electronics Society Press, 2003
- B. Daum, U. Merten, *System Architecture with XML*, Elsevier Science, 2003
- J. Davies *et al.* (eds.), *Towards the Semantic Web*, John Wiley & Sons, 2003
- V. Geroimenko, *Dictionary of XML Technologies and the Semantic Web*, Springer, 2004
- D. Jackson (ed.), *Scalable Vector Graphics (SVG) 1.2*, W3C Working Draft, Boston, 2004: <http://www.w3.org/TR/SVG12/>
- N. Mitra (ed.), *SOAP Version 1.2 Part 0: Primer*, W3C Recommendation, Boston, 2003: <http://www.w3.org/TR/soap12-part0/>
- N. Si Shi, V.K. Murthy, *Architectural Issues of Web-Enabled Electronic Business*, Idea Group Publishing, 2003
- M. Smith *et al.* (eds.), *OWL Web Ontology Language Guide*, W3C Recommendation, Boston, 2004: <http://www.w3.org/TR/owl-guide/>
- ***, *ABC Modeling Examples*: http://metadata.net/harmony/cimi_modelling.htm
- ***, *FOAF (Friend Of A Friend) Specification*: <http://rdfweb.org/>
- ***, *OWL-S Specification*: <http://www.daml.org/services>
- ***, *World-Wide Web Consortium*, Boston, 2005: <http://www.w3.org/TR/>