Cardoso, J., "Integrating HAD Organizational Data Assets using Semantic Web Technologies", 3rd International Conference Interoperability for Enterprise Software and Applications (I-ESA 2007), Funchal, Portugal, March 28th-30th, 2007. Enterprise Interoperability II, Ricardo Gonçalves, Jörg Müller, Kai Mertins, and Martin Zelm (eds), pp. 333-344, Springer. ISBN: 978-1-84628-857-9.

Integrating HAD Organizational Data Assets using Semantic Web Technologies

J. Cardoso
Department of Mathematics and Engineering, University of Madeira, 9050-390 Funchal, Portugal.
jcardoso@uma.pt

Keywords: Semantic Web, Data Integration, Data Semantics, Enterprise Data Management, Semantic Information Architecture, XML, OWL.

ABSTRACT

To create value based on information and knowledge, organizations need to recognize that they are composed of various types of data assets. In this context, data integration has been proven to be a challenge due to the heterogeneity of the information systems involved. The Extensible Markup Language (XML) was one of most successful solutions developed to provide business-to-business integration. However, XML "lacks semantics". Thus problems arise when it is necessary to manipulate and integrate different XML data sources. Consequently, today's organizations are again shifting from a syntactic interoperability level to a semantic one. This paper presents a Semantic data Integration Middleware (SIM) which, when based on a single query, integrates data residing in different data sources. The middleware uses an ontology-based multi-source data extractor/wrapper approach to transform data into semantic knowledge.

1 Introduction

Barnett and Standing (Barnett and Standing, 2001) argue that the rapidly changing business environment brought on by the Internet requires organizations to implement new business models as rapidly, develop new networks and alliances, and be creative in their marketing strategies. In order to compete in the electronic

era, businesses must be prepared to use technology-mediated channels, create internal and external value, formulate technology convergent strategies, and organize resources around knowledge and relationships (Rayport and Jaworski, 2001).

To create internal and external value based on information and to organize resources around knowledge and relationships, organizations need to recognize that they are composed of various types of data assets. Examples of data asset formats include relational databases, Web pages, plain text files, EDI documents (Electronic Data Interchange), XML files, and, more recently, Web services. Despite the large quantity of data collected by organizations, managers often struggle to obtain information that would help them in decision-making. In this context, data exchange and integration has been proven to be a challenge due to the heterogeneity of the information systems involved.

At least three types of data heterogeneity may occur when integrating information from heterogeneous, autonomous, and distributed (HAD) data schema (Sheth, 1998, Ouskel and Sheth, 1999): syntactic heterogeneity: the technology supporting the data sources differs (e.g. databases, Web pages, XML streams, Web services, etc); schematic heterogeneity: data sources schema have different structures; and semantic heterogeneity. These heterogeneity problems create a market for the creation and the maintenance of point-to-point translations between these schemas worth billions dollars per annum (Schreiber, 2003). When these translations are carried out manually they are expensive to create. Moreover, since most of the time they are not based on the semantic understanding of the data, they result in poor information integration quality. In 2003, the cost of this limitation was estimated to be $600 billion/year for the US (Eckerson, 2003).

The need to integrate of heterogeneous, autonomous, and distributed (HAD) data and systems has become a hard task since several data representations, formats and schema exist nowadays. Figure 1 illustrates the evolution of the various data representations over the years. Several decades ago, organizations stored their data in static flat files. However, in answer to the inherent dynamic nature of businesses, organizations started to rely on dynamic solutions to manage their data. One solution was the worldwide adoption relational database management systems (RDBMS). This technique offers the opportunity to deliver information that is highly customized to the needs of individual users.

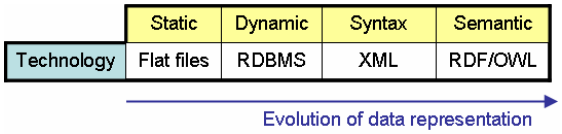

Fig. 1. Evolution of data representations

Nevertheless, the technologies available to query data based on databases were insufficient for the requirements of organizations looking for application integration solutions. Businesses required their heterogeneous systems and applications to communicate in a transactional manner. The Extensible Markup Language (XML, 2005) was one of most successful solutions developed to provide

business-to-business integration. XML became a means of transmitting unstructured, semi-structured, and even structured data between systems, enhancing the integration of applications and businesses.

XML brought syntactic interoperability and became the *de facto* standard as a B2B data exchange format (Bussler, 2003). However, XML "lacks semantics" (Hawke, 2001, Shabo A. et al., 2006). Thus problems arise when it is necessary to manipulate and integrate different XML data sources. Consequently, today's organizations are again shifting (or it is expected of them to do so) from a syntactic interoperability level to a semantic one (EBizQ, 2005).

Approaches to the problems of semantic heterogeneity should equip heterogeneous, autonomous, and distributed software systems with the ability to share and exchange information in a semantically consistent way (Sheth, 1998). A suitable solution to the problem of semantic heterogeneity is to rely on the technological foundations of the semantic Web; or more precisely, to semantically define the meaning of the terminology of each distributed system data using the concepts present in a shared ontology to make clear the relationships and differences between concepts.

Schreiber (Schreiber, 2003) points out that the Semantic Web and its underlying technologies may have their greatest impact within organizations that struggle with business information being spread across thousands of data sources each of which is semantically different. Furthermore, it has been identified that the Semantic Web has already demonstrated its practical use in Bioinformatics, Web services, Tourism Information Systems, Digital Libraries, etc (Cardoso and Sheth, 2006, Cardoso and Sheth, 2005).

This paper presents a Semantic data Integration Middleware (SIM) which, when based on a single query, integrates data residing in different data sources possibly with different formats, structures, schema, and semantics. The middleware uses an ontology-based multi-source data extractor/wrapper approach to transform data into semantic knowledge (Silva and Cardoso, 2006). SIM is composed of two main modules: the Schematic Transformation module and the Syntactic-to-Semantic Transformation module. The first module is responsible for integrating data residing in different data sources possibly with different formats, structures, and schema. The second module maps XML Schema documents to existing OWL ontologies and automatically transform XML instance documents into individuals of the mapped ontology (Rodrigues et al., 2006). This module is crucial for organizations that plan to move from a syntactic representation of data using XML to a semantic one using OWL.

This paper is structured as follows. In Section 2, the architecture of our system, namely, semantic data integration middleware (SIM) is presented . In Section 3, the Schematic Transformation module is described. This module uses a multi-source data extractor/wrapper approach to transform data to an XML representation. In Section 4 the second most important module of SIM is described, the Syntactic-to-Semantic Transformation module which automatically transforms XML instance documents into individuals of an OWL ontology. In Section 5 the related work in data integration middleware is presented and finally, in section 6, our conclusions.

2 SIM Architecture

The development of our Semantic data Integration Middlware (SIM) is a complex issue since it requires the integration of distributed systems with infrastructures that are not frequently encountered in more traditional centralized systems. For a SIM to be successful it is indispensable to study its architecture. The study of architectural strategies has a critical impact on early decisions in system development; it is both cost-effective and efficient to conduct analyses at the architecture level, before substantial resources have been committed to development (Bass et al., 1998). Therefore, we will undertake a study of our approach to SIM development by presenting its architecture.

We propose architecture for SIM composed of four layers: data sources, Schematic Transformation, Syntactic-to-Semantic Transformation, and ontology. The relationships between these layers are illustrated in Figure 2.

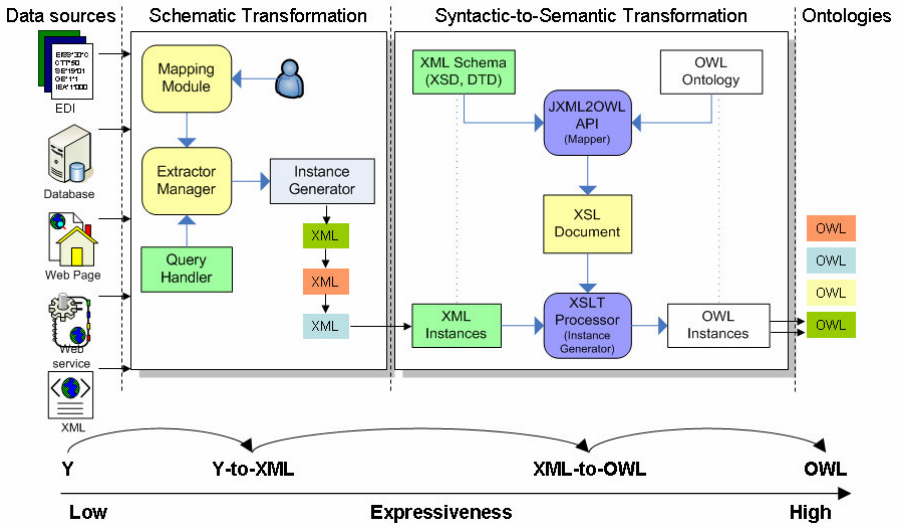

Fig. 2. Overview of SIM architecture

These four modules have the following objectives and responsibilities:

Data Sources (Y): The data sources define the scope of the integration system, thus data source diversity provides a wider integration range and data visibility. SIM can connect to B2B traditional data source formats, such as structured (e.g. relational databases), semi-structured (e.g. XML) and unstructured (e.g. Web pages and plain text files), EDI (Electronic Data Interchange), and Web services. The supported data source types can easily be increased to support other formats.

Schematic Transformation (Y-to-XML): This module integrates data residing in different data sources possibly with different formats, structures, schema, and

semantics. The module uses a multi-source data extractor/wrapper approach to transform data to a XML representation.

Syntactic-to-Semantic Transformation (XML-to-OWL): This module uses the JXML2OWL framework (described in section 4.2) to map XML Schema documents to existing OWL ontologies and automatically transform XML instance documents into individuals of the mapped ontology. This module is crucial for organizations that plan to move from a syntactic representation of data using XML to a semantic one using OWL.

Ontologies (OWL): SIM introduces the ability to extract data from various data source types (unstructured, semi-structured, and structured) and wrap the result in OWL (Web Ontology Language) format (OWL, 2004), providing a homogenous access to a heterogeneous set of information sources. The decision to adopt OWL as the ontology language is based on the fact that this is the World Wide Web Consortium (W3C) recommendation for building ontologies.

Since the two most important components of our middleware are the Schematic Transformation and Syntactic-to-Semantic Transformation modules, they will be described in detail in the next two sections.

3 Schematic Transformation

As organizations grow and change, their needs to manage and access information increases exponentially. In many situations, "data supporting architectures have shifted from a centralized to a distributed approach due to the advantages in the cost and flexibility". While these trends have resulted in many advantages for organizations, they have also introduced a large gap in the ability to integrate data between applications and organizations.

A middleware for data integration should allow users to focus on *what* information is needed and leave the details on *how* to obtain and integrate information hidden from users (Silva and Cardoso, 2006). Thus, in general, data integration systems must provide mechanisms to communicate with an autonomous data source, handle queries across heterogeneous data sources, and combine the results in an interoperable format. Therefore, the key problem is to bridge syntactic, schematic and semantic gaps between data sources, thereby solving data source heterogeneity.

At least three types of data heterogeneity may occur when integrating information from heterogeneous, autonomous, and distributed data sources: syntactic heterogeneity: the technology supporting the data sources differs (e.g. databases, Web pages, XML streams, Web services, etc); schematic heterogeneity: data sources schema have different structures; and semantic heterogeneity: data sources use different meanings, nomenclatures, vocabulary or units for concept.

The Schematic Transformation module is responsible for integrating data from different data source and resolving syntactic heterogeneity and schematic

heterogeneity. Semantic heterogeneity is solved by the Syntactic-to-Semantic Transformation module.

3.1 Architecture

Figure 2 presents a high level illustration of the Schematic Transformation module. Two key areas can be identified. The first concerns the extractor (Extractor Manager) used to connect to the different data sources registered in the system and to extract data from them. The extracted data fragments are then compiled in order to generate ontology instances. The second key area is the mapping result between an ontology schema and the data sources (Mapping Module). This information is produced when the ontology attributes and classes are intersected with the data sources forming an extraction schema used by the extractor to retrieve data from the sources.

Other areas also play an important role in the architecture. This is the case of the Query Handler, which receives and handles the queries to the data sources, the Instance Generator, which is responsible for providing information about any error that has occurred during the extraction process or in the query, and finally the Ontology Schema that plays a major role in data mapping.

3.2 Mapping Module

To enable the extraction from distributed and heterogeneous sources, it is necessary to formally denote the notion of mapping between remote data and the local ontology. The mapping is the result of information crossing between the XML schema and the data sources in order to provide information about XML's attributes in the extraction process.

Depending on data source characteristics, two data extraction scenarios may emerge. This is because data sources might have 'one' data record (for instance a Web page describing a watch) or might have 'n' data records (for instance a database of watches). The data source scenario defines how the mapping is made and how data is extracted (in order to support the existence of an infinite number of records).

According to our approach, the mapping procedures are carried out manually. This task is time consuming but offers the highest degree of data extraction accuracy and domain consistency. This fact is very important when integrating data since the integrity and correlation between the sources and the schema must be very accurate so that the "meaning" of the data is not lost. Although time consuming, the mapping should not need substantial maintenance after being created. Data sources do not normally change their structures (except perhaps Web pages), so few mapping updates should be necessary.

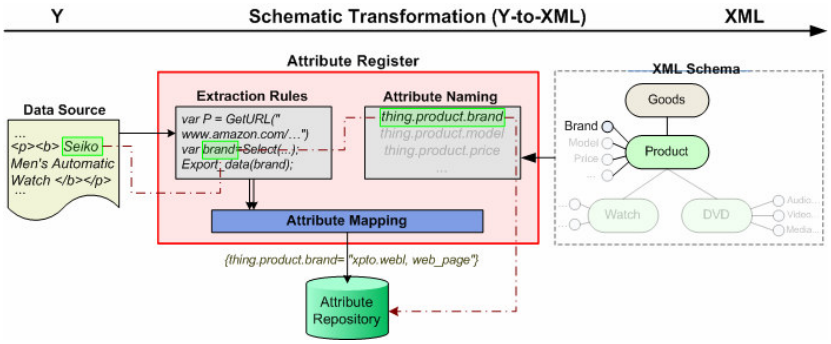

Fig. 3. Attribute Registration

In order to register an attribute we need information about the XML schema and how to extract the information from a specific data source. The objective is to have a mapping specification that relates information about attributes, data sources and extraction rules.

Figure 3 illustrates the attribute registration process. In the example the data source is a Web page, so the extraction rules were set using a Web extraction language. The attribute registration process requires a set of steps to be completed in order to achieve a correct mapping. The first step is to name the attributes. The second step is to define the extraction rules. The last step maps the attribute with the extraction rule.

3.2 Extractor Manager

This component handles data sources for retrieving the raw data to accomplish query requirements. The extraction method varies by data source so the extractor must support several extraction methods. The extractor and mapping architecture were designed in order to be easily extended to support other extraction methods and languages.

This is the main section of the Schematic Transformation module and it is implemented by three tasks, Obtain Extraction Schema, Obtain Data Source Definition and Data Extraction. After processing the query, the system must retrieve data in order to answer the query. The extraction is based on attributes, so this area retrieves extraction schemas of the required attributes, thus indicating to the extractor how the extraction is executed.

Attributes are associated with data sources and data sources have connection characteristics. Therefore, extractors need to know how to connect to each data source. After retrieving an extraction schema, the extractor fetches the associated data source definition to enable its access. Now extraction can take place. This is the hot point in the extraction mechanism. It is supported by a mediator and a set of wrappers/extractors (details will be given in the subsequent sections).

3.3 Instance generator

This module serializes the output data format and handles the errors from the queries and from the extraction phases. The Schematic Transformation module transforms structured, semi structured or unstructured formats to XML. The population process (XML instance generation) is executed in an automatic way. This is because the extracted information respects the XML schema.

3.3 Query Handler

A query is the event that sets the Schematic Transformation module in action. The input is based on a higher level semantic query language. This query is then transformed to represent requests based on XML elements. The Syntactic-to-Semantic Query Language (S2SQL) is the query language based on SQL supported by the extraction module. It is a simpler version of SQL since data location is transparent from the query point of view. Thus the FROM and related operators have no use in S2SQL and are thus not supported. This way, queries are created only with the indication of which data is required. It is not necessary to supply information about data location, data format, extraction method, etc.

4 Syntactic-to-Semantic Transformation

Today's enterprises face critical needs in integrating disparate information spread over several data sources inside and even outside the organization. Most organizations already rely on XML standards to define their data models. Unfortunately, even when using XML to represent data, problems arise when it is necessary to integrate different data sources. Emerging Semantic Web technologies, such as ontologies, RDF, RDFS, and OWL, can play an important role in the semantic definition and integration of data. The Syntactic-to-Semantic Transformation module allows organizations to move from a syntactic data infrastructure defined in XML to a semantic data infrastructure using OWL. The module supports mappings and fully automated instance transformation from syntactic data sources in XML format to a common shared global model defined by an ontology using Semantic Web technologies. This module allows organizations to automatically convert their XML data sources to a semantic model defined in OWL.

4.1 Semantic Model

To conceptualize a domain in a machine readable format an ontology is necessary. In B2B applications, ontologies play an important role in order to promote and facilitate interoperability among systems, to enable intelligent processing, and to share and reuse knowledge. From a data integration point of view, ontologies provide a shared common understanding of a domain.

SIM represents ontologies using the Web Ontology Language (OWL), a semantic markup language for publishing and sharing ontologies on the World

Wide Web. Other alternative formal languages can also be used to express ontologies, for instance CycL (Cycorp, 2006), KIF (Genesereth, 2006) and RDF (Lassila and Swick, 1999).

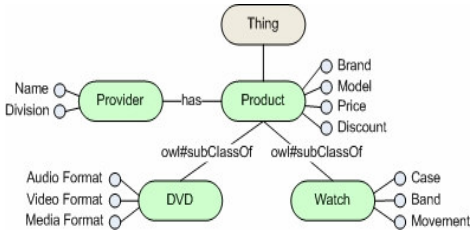

Fig. 4. Ontology schema example

Since the ontology schema defines the structure and the semantics of data (Figure 4) it is understandable that there is a need for the schema in the extraction process. The ontology is used to create mappings between data sources and the schema. Another important role of the ontology schema is to define the query specification process.

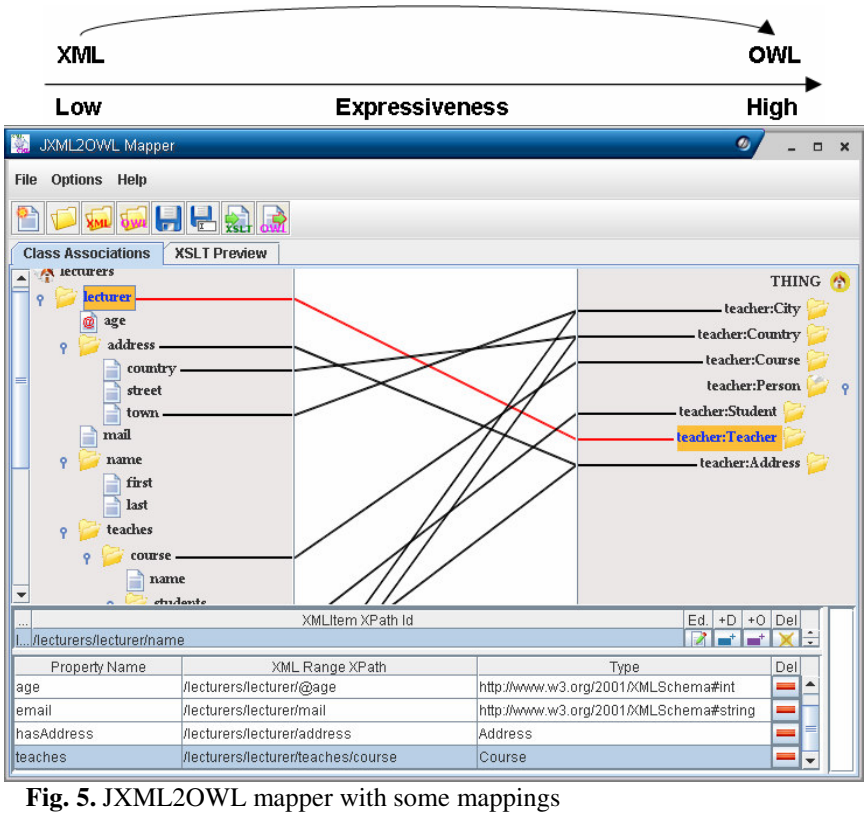

Fig. 5. JXML2OWL mapper with some mappings

4.2 JXML2OWL framework

JXML2OWL (Rodrigues et al., 2006) is a framework divided in two sub projects: JXML2OWL API and JXML2OWL Mapper. The API is a generic and reusable open source library for mapping XML schemas to OWL ontologies for the Java platform while the Mapper is an application with a graphical user interface (GUI) developed in Java Swing that uses the API and eases the mapping process (Figure 5).

JXML2OWL supports manual mappings from XML, XSD or DTD documents to an OWL ontology, thus supporting all the kinds of mappings such as many-to-many. Currently, conditional mappings through XPath predicates are not implemented within the framework. According to the mapping performed, JXML2OWL generates mapping rules wrapped in an XSL document that allows the automatic transformation of any XML data, that is, any XML document validating against the mapped schema, into instances of the mapped ontology. Figure 2 represents such process in the Syntactic-to-Semantic transformation module.

With JXML2OWL, the mapping process requires several steps. The first step consists of creating a new mapping project and loading both the XML Schema related file (XSD or DTD) and the OWL ontology. If an XML schema in not available, it is possible to load an XML document. In this case, JXML2OWL extracts a possible schema. In the second step, the user creates class mapping between elements of the loaded XML schema and classes of the ontology. Once these mappings are created, it is possible to relate them to each other to create object property mappings, or to relate them with elements of the XML schema to create data type property mappings. Finally, in the last step, it is possible to export the transformation rules, generated according to the mapping performed, as an XSL document. With this XSL document it is possible to transform any XML document which validates against the mapped XML schema into individuals of the mapped OWL ontology. Obviously, both the API and the Mapper support all these steps. The produced OWL instances document can evidently be loaded in any OWL editor such as Protégé-OWL (Protégé, 2005).

5. Related work

There are several research projects which target the same objectives as the SIM middleware. The main differences are that we use semantics and ontologies to achieve a higher degree of integration and interoperability. The World Wide Web Wrapper Factory (W4F) (Sahuguet and Azavant, 1999) toolkit is a good framework to develop Web wrapper/extractor. It allows the user to create Web wrappers and deploy them as modules in a bigger application. W4F extracts data. exclusively from Web pages and the output may be in an XML file or a Java interface. The Caméléon Web Wrapper Engine (Firat et al., 2000) is capable of extracting from both text and binary formats. The engine provides output in XML. Artequakt (Alani et al., 2003) is an Automatic Ontology-Based Knowledge Extraction from Web documents that automatically extracts knowledge from an

artistic ontology and generates personalized biographies. The major drawback of this system is that it is customized to a specific domain. The Architecture for Semantic Data Access to Heterogeneous Information Sources (Rishe et al., 2000) allows heterogeneous data sources to have uniform access through a common query interface based on a Semantic Data Model.

6. Conclusions

Semantic integration of HAD data assets is one of the most difficult and costly tasks in enterprise information technology. The Semantic Web can help in the integration of multiple physical heterogeneous data schema by mapping schema to one, or more ontologies which reflects the desired business world-view. Therefore, in this paper we have presented middleware architecture (called SIM) to semantically integrate organizational data assets. The main goal of the architecture is to offer a common understanding of a domain and assimilate heterogeneous systems using Semantic Web technology. All this is supported by an ontology schema thus offering semantic data representation benefits that allow data to be shared and processed by automated tools as well as by people.

SIM supports organizational data assets represented in various data storage and data message formats such as flat files, EDI documents, XML, relational databases, etc. To achieve integration, SIM transforms these schema using extractors/wrappers and syntactic mappings to infer translation scripts between the data assets and an intermediate XML data representation. Once the data assets are stored using an XML schema, the Syntactic-to-Semantic Transformation module maps XML documents to existing OWL ontologies and automatically transform XML instances documents into individuals of the mapped ontology. Such framework is crucial for organizations that plan to move from a syntactic representation of data using XML to a semantic one using OWL.

SIM has been successfully employed to integrate disparate e-tourism data sources as individuals of an e-tourism OWL ontology.

Acknowledgement. This work was funded by grants from the FCT (Fundação para a Ciência e a Tecnologia) and carried out in cooperation with Expedita. We would like to express recognition of Toni Rodrigues, Pedro Rosa, and Bruno Silva for the implementation of the SIM system.

7 References

Alani, H., Kim, S., Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H. and Shadbolt, N. R. (2003) *IEEE Intelligent Systems,* **18,** 14-21.

Barnett, M. and Standing, C. (2001) *Journal of Vacation Marketing,* **7,** 143-152.

Bass, L., Clements, P. and Kazman, R. (1998) *Software Architecture in Practice,* Addison Wesley.

Bussler, C. (2003) *B2B Integration: Concepts and Architecture,* Springer-Verlag.

Cardoso, J. and Sheth, A. (2005) *Semantic Web Process: powering next generation of processes with Semantics and Web services,* Lecture Notes in Computer Science, Springer-Verlag, Vol. 3387, ISBN:3-540-24328-3, Heidelberg.
Cardoso, J. and Sheth, A. (2006) *Semantic Web Services, Processes and Applications,* Springer, ISBN:0-38730239-5.
Cycorp (2006) Cyc KNowledge Base - http://www.cyc.com/.
EBizQ (2005) Semantic Integration: A New Approach to an Old Problem, http://www.ebizq.net/white_papers/5988.html Software AG.
Eckerson, W. (2003) Data Quality and the Bottom Line, Vol. 2003 The Data Warehousing Institute. http://www.dw-institute.com/research/display.asp?id=6028.
Firat, A., Madnick, S. and Siegel, M. (2000) The Caméléon Web Wrapper Engine Cairo, Egypt, pp. 269-283.
Genesereth, M. (2006) Knowledge Interchange Format (KIF) - http://logic.stanford.edu/kif/dpans.html.
Hawke, S. (2001) XML with Relational Semantics: Bridging the Gap to RDF and the Semantic Web, http://www.w3.org/2001/05/xmlrs/. Vol. 2001 W3C.
Lassila, O. and Swick, R. (1999) Resource Description Framework (RDF) model and syntax specification. W3C Working Draft WD-rdf-syntax-19981008. http://www.w3.org/TR/WD-rdf-syntax.
Ouskel, A. M. and Sheth, A. (1999) *SIGMOD Record,* **28,** 5-12.
OWL (2004) OWL Web Ontology Language Reference, W3C Recommendation, Vol. 2004 World Wide Web Consortium, http://www.w3.org/TR/owl-ref/.
Protégé (2005) Protégé, Vol. 2005 Stanford Medical Informatics.
Rayport, J. F. and Jaworski, B. J. (2001) *e-Commerce,* McGraw-Hill, Boston.
Rishe, N., Vaschillo, A., Vasilevsky, D., Shaposhnikov, A. and Chen, S.-C. (2000) The Architecture for Semantic Data Access to Heterogeneous Information Sources New Orleans, Louisiana, USA, pp. 134-139.
Rodrigues, T., Rosa, P. and Cardoso, J. (2006) Moving from Syntactic to Semantic Organizations using JXML2OWL, Department of Mathematics and Engineering, University of Madeira, Report nº20061, Funchal, Portugal.
Sahuguet, A. and Azavant, F. (1999) Building Intelligent Web Applications Using Lightweight Wrappers Edinburgh, Scotland, UK, pp. 738-741.
Schreiber, Z. (2003) Applying the Semantic Web Vision to Enterprise Data Management: A Case Study. Budapest, Hungary, pp. 79.
Shabo A., S., R.-C. and P., V. (2006) *IBM Systems Journal,* **45,** 361-372.
Sheth, A. (1998) In *Interoperating Geographic Information Systems*(Eds, Goodchild, M. F., Egenhofer, M. J., Fegeas, R. and Kottman, C. A.) Kluwer, Academic Publishers, pp. 5-30.
Silva, B. and Cardoso, J. (2006) Semantic Data Extraction for B2B Integration IEEE Computer Society, Lisboa, Portugal.
XML (2005) Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004 http://www.w3.org/TR/REC-xml/.