

# AN ARCHITECTURE FOR INTEROPERABILITY OF ENTERPRISE INFORMATION SYSTEMS BASED ON SOA AND SEMANTIC WEB TECHNOLOGIES

Fuqi Song, Gregory Zacharewicz and David Chen

*IMS CNRS UMR 5218 / GRAI, Université de Bordeaux, 351, cours de la Liberation, 33405 Talence Cedex, France*

**Keywords:** Enterprise information systems interoperability, Semantic interoperability, Semantic web, SOA, Ontology.

**Abstract:** Enterprises interoperability becomes a necessary and critical task for enterprises as the information and complexity of systems increase exponentially. To adapt existing information systems to new requirements and build bridges between them, a conceptual vision and understanding of interoperability can facilitate the technical development to reduce the conflicts and gaps between heterogeneous systems. Now the focus shifts from the syntactic concern to semantic issues. Semantic heterogeneity currently becomes the next barrier and challenge to face in enterprise interoperability, it promises to play a major role to enable interoperability of enterprise information systems. The approach can be transposed at enterprise decision level, where heterogeneous business processes also need to be interoperable. SOA helps to explore a loosely coupled architecture with many advantages, especially for enhancing interoperability among enterprises. To obtain business process interoperability, service orchestration and choreography could coordinate and compose the services and processes effectively. This paper, first gives a brief introduction of main issues in this domain, and then recalls a framework for enterprise interoperability. Based on these methodologies and the interoperability framework, a new architecture for enterprise interoperability of information systems using SOA ideology and semantic web technologies is proposed, whose final objective is to enable and enhance enterprise interoperability and then create business values.

## 1 INTRODUCTION

The development of information systems is a key issue which influences the progress of enterprises nowadays. Since the information systems have been applied for decades in enterprises, numerous information systems had been developed in various domains and industries, so they are coexisting at the same time independently. However isolated information systems cannot adapt the rapid business changes, such as the globalization of production, economic crisis, marketing changes, etc. Thereby new demands that came up requires multiple information systems work together to undertake the tasks that one single system could not complete. This is the problem of interoperability, whose goal is to fulfill the gap between new business requirements and existing information systems in enterprises.

The main basic barrier, which prevents to achieve interoperability, is the heterogeneity, both structural and semantic. Data integration is a

solution to solve the problem of information system interoperability before. It mainly solves the structural heterogeneity, because these systems were relatively simple. However, as the amount and complexity of information systems increase, the structural heterogeneity becomes a less important issue, instead, the semantic heterogeneity becomes new challenge and focus.

Besides the semantics issues, the issue of business processes among different information systems is crucial for enterprises. Because the ultimate goal of enabling information system interoperability is to make business interoperable. The objective of business process interoperability is to coordinate and compose the services into process to fulfill new and more complex business requirements.

Thus semantic and business process issues are two significant points for constructing a flexible and effective architecture for information systems interoperability. This paper aims to propose an operational architecture to facilitate interoperability

of enterprise information systems from IT and enterprise modeling perspectives. On one hand, with applying SOA ideology and methodology it provides a loosely coupled architecture and enhance the business process integration. On the other hand, the use of semantic web technologies allows solving semantic heterogeneity.

### 1.1 Framework for Enterprise Interoperability (FEI)

Enterprise interoperability is the ability for two systems to understand one another and to use functionality of one another. In enterprises, solutions need to be implemented from multiple aspects to facilitate interoperability. In (Chen, 2006, 2008) a three dimensional framework (see figure 1) has been proposed to represent the ability of interactions between enterprises systems. First the framework identifies the barriers which are obstacles for achieving interoperability among enterprises: conceptual, technological and organizational. Then four concern points are defined: data, service, process and business, these points should be taken into account when architectures and solutions are built. Generic approaches used for enterprise interoperability are categorized into: integrated, unified and federated.

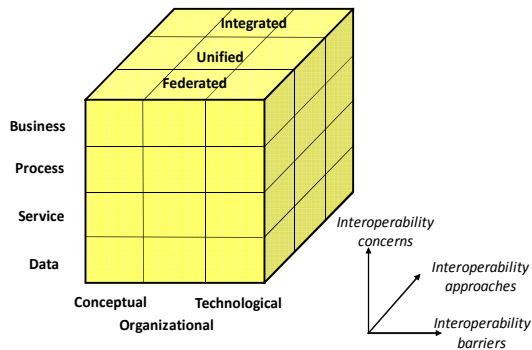


Figure 1: Framework for enterprise interoperability.

### 1.2 Service-Oriented Architecture (SOA)

Service-oriented architecture (SOA) is a flexible set of design principles used during the phases of systems development and integration in computing. A system based on a SOA will package functionality as a suite of interoperable services that can be used within multiple separate systems from several business domains. Traditional methods are highly coupled, this doesn't facilitate the enterprise interoperability, and is not flexible for scalability,

extendibility and maintenance. Service-orientation requires loose coupling of services with operating systems, and other technologies that underlie applications. SOA separates functions into distinct units, or services, which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications (Bell, 2008).

A service is usually defined as the realization of business functionality via software that anyone can use, anywhere, to compose new applications by using their services in the context of new or modified processes (Izza, 2008). The goal of the web services effort is to achieve interoperability between applications by using Web standards. Web services use a loosely coupled integration model to allow flexible interaction of heterogeneous systems in a variety of domains including business-to-consumer, business-to-business and enterprise application.

Service orchestration and choreography are two terms used to describe the composition and collaboration of services. A service is a unit of functionality, which could be regarded as atomic to provide users certain functions. Most of time, these services are independent with each other, this may cause the waste and redundancy, because one service may be reused for different business processes.

Service orchestration refers to an executable business process that may interact with both internal and external web services, whereas service choreograph means more collaborative in nature, where each party involved in the process describes the part they play in the interaction. Orchestration differs from choreography in that it describes a process flow between services, controlled by a single part. More collaborative in nature (see figure 2), choreography tracks the sequence of messages involving multiple parties, where no one party truly "owns" the conversation (Peltz, 2003).

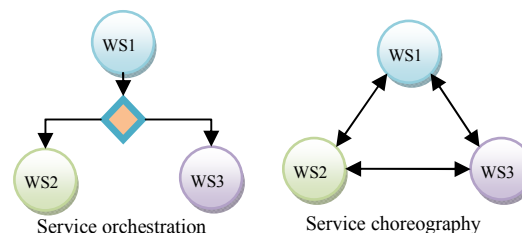


Figure 2: Web services orchestration and choreography.

The following of the paper is organized as; section 2 analyzes the semantic issues in information systems interoperability. Section 3 describes the proposed architecture and elaborates each layer. In section 4, we discuss the proposed architecture compared with some existing approaches. Last

section presents some future work that will be done to this architecture.

## 2 SEMANTIC ISSUES

The methods for data interoperability have a trend shifting from structural to semantic (Ziegler, 2004). Because the volume of information increase exponentially and more systems get involved, to find useful information and retrieve them from the huge amount of data becomes necessary. The traditional structure-oriented approaches have limitations for retrieving “meaningful” information, because of insufficient emphasis on semantic aspects.

### 2.1 Semantic Heterogeneity

(Buccella, 2003) classify heterogeneity into four categories: (1) structural heterogeneity, involving different data models; (2) syntactical heterogeneity presents different languages and data representations; (3) systemic heterogeneity, involving hardware and operating systems; and (4) semantics heterogeneity involves different concepts and their interpretations. In general, the semantic heterogeneity deals with three types of concepts: the semantically equivalent concepts, the semantically unrelated concepts, and the semantically related concepts. In the first case, different models use different terms to refer the same concept, for example, one system may use “teacher” whereas the other uses “professor”. In the second case, the same term may be used by different systems to denote completely different concepts, such as “dear” may refer to “expensive” or address a person; and in the last case, different classifications may be performed, for example one system classifies “student” as “male” and “female” and other system as “bachelor” and “master”.

### 2.2 Ontology

Ontology is regarded as the main approach to solve the semantic problems. A classical definition considers ontology as a specification of a conceptualization (Gruber, 1993). Ontology can represent the meaning of vocabularies themselves and their relations. Ontology is frequently used in data integration for solving the content explication. In (Wache, 2001), many approaches for information integration based on ontology are surveyed, and three possible modes are concluded: single ontology, multiple ontologies and hybrid approaches. Each mode has its constraints and benefits, single

ontology is easy to implement but is not adaptive when adding or removing a data source, whereas multiple ontology implement costly but has more flexibility with data source changes. Hybrid mode facilitates comparison among different sources due to the shared vocabulary but hard to maintain.

### 2.3 Semantic Web

The semantic web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. The ultimate goal of the Web of data is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network (W3C, 2010). Semantic web is a group of methods and technologies to enable the construction of a connected and reusable data network. (Bratt, 2007) presented a stack to illustrate the main semantic web technologies.

## 3 A SOA AND SEMANTIC WEB TECHNOLOGIES BASED APPROACH

The objective of our approach is to propose a loosely coupled architecture for information systems interoperability. To obtain high interoperability, we explore the techniques and methods that could facilitate and benefit it, thereby SOA and semantic web technologies are applied to design this architecture. The idea is not developing a new dependant and separate information system, but establishing a transparent layer among enterprises information systems so that they can share and exchange information without semantic barriers.

### 3.1 Adaption to the Framework

Based on the enterprise interoperability framework and the approaches used in (Zacharewicz, 2009), we focus in the framework to a specific scope: information system interoperability. We consider all the barriers and three concern points: data, service and process. A federated approach is applied due to the setup of a semantic layer. The mapping is shown in figure 3 to situate the proposed architecture in the Framework.

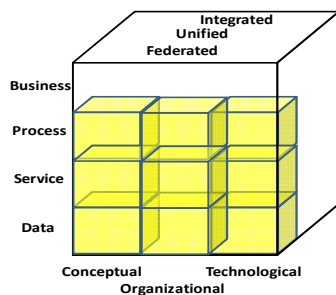


Figure 3: Position of the proposed architecture in the framework.

### 3.2 Overview

The architecture can be applied in one enterprise or among several enterprises (ex: partner relation, collaborations) to enable business corporation among them via their own information systems, and keep their own applications independent. The architecture includes five layers as illustrated in figure 4. Each layer is independent to the others, which communicate through defined interfaces and protocol. The lower levels are more concerned with data, in the contrast, the higher level focuses more on business and users.

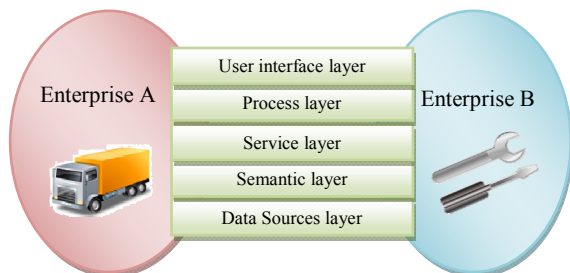


Figure 4: Overview of the architecture.

In short, the data interoperability part is carried out in the two lower layers, and the service and business interoperability part is done in the two upper levels, the highest level deals with user interaction. The architecture covers three interoperability concerns: data, service and processes. A semantic layer upon data sources is designed to emphasize the semantic heterogeneity of information, not only the information semantics, the semantic web services are also built in the service layer to tackle conceptual interoperability barriers. In addition, the SOA-based methods are used for solving the technological barriers and providing a federated architecture. The business coordination and collaboration take place at process layer, and get end users are involved via the user interface layer, to remove the organizational barriers.

First, a semantic layer is built upon data sources, which is equivalent to the information in the data sources, through a rule-based processor to map between the semantic layer and data sources. It does not aim to transform the data from the data sources to the ontology server, but a mapping connection built between them. When retrieving or updating the data, the real action needs to be effected on the data sources through the semantic layer. And a monitor component is set up to keep semantic adaptive and evolutionary when the information and semantic change.

Then the upper level is service layer, in which the services are regarded as atomic. The services layer is core part of the architecture, because it provides functionalities and connects information and business processes. Namely, it retrieves information from lower level to implement the services, and then provides the atomic services to compose business process. Compared with the single services in this layer, the business process layer is linked to the real business process performed in enterprises. The processes are composed by the atomic services in services layer. In order to integrate the services into process, services orchestration and choreography are applied for coordinating or composing services. In particular, the description languages WS-BPEL (Web Services Business Process Execution Language) and WS-CDL (Web Services Choreography Description Language) are used for service orchestration and choreography respectively.

The highest level is the user interface level, which the user interacts with directly, in this layer the systems are the ones existed in enterprises or the future system to be developed. It connects to the enterprise service bus (ESB) with business process layer and service layer, in order to call the services to accomplish the business processes.

### 3.3 Elaborations of Each Layer

In this section, we define each layer of the architecture; the detailed illustration is shown in figure 5.

#### 3.3.1 Data Sources Layer

This layer represents the data sources located in different enterprises and domains, They can be accessed directly or through the interfaces they provide. These sources have three characteristics: heterogeneous, distributed and autonomous. The details of each characteristic are as follows.

Heterogeneous: They have different structure and format, the most common storage method is rational

database, which is a well-structured format. Another one is file. However, the formats of files vary a lot depending on its property. For example HTML/XML/XHTML is well-structured sources, whereas excel sheet, word document are semi-structured sources. As well as non-structured sources, such as plain text document (.txt) and some other documents with special designed format.

**Distributed:** The sources locate on different domains either physically or logically. Such as in one enterprise, the database could be deployed to several cities, because there is one branch in each city. Even two databases are deployed in one same server, they belong to different database system, for example, Oracle and DB2 respectively, this is an instance of logically distributed.

**Autonomous:** Each source is logically autonomous, each source can be regarded as one system, and they are dependent to each other. They are connected to the systems via interfaces or API (Application Programming Interface).

### 3.3.2 Semantic Layer

Upon the structural data sources, a semantic layer is established to represent the semantics of the information. Not all information will be taken into account when building the semantic layer, but only the information which is useful and valuable to the integrated business. Thus a rule-based processor is constructed to map and retrieve data from the lower data sources layer. With defined rules, the processor could filter and control the data so as to build a concise and effective semantic layer. The processor can also update the data sources when a change request is sent from the upper level.

Ontology is used for representing the semantics of information. An ontology server is deployed as the storage of the ontology. As we introduced in the introduction in the recall section, the ontology server is applied with the global schema approach due to its advantages.

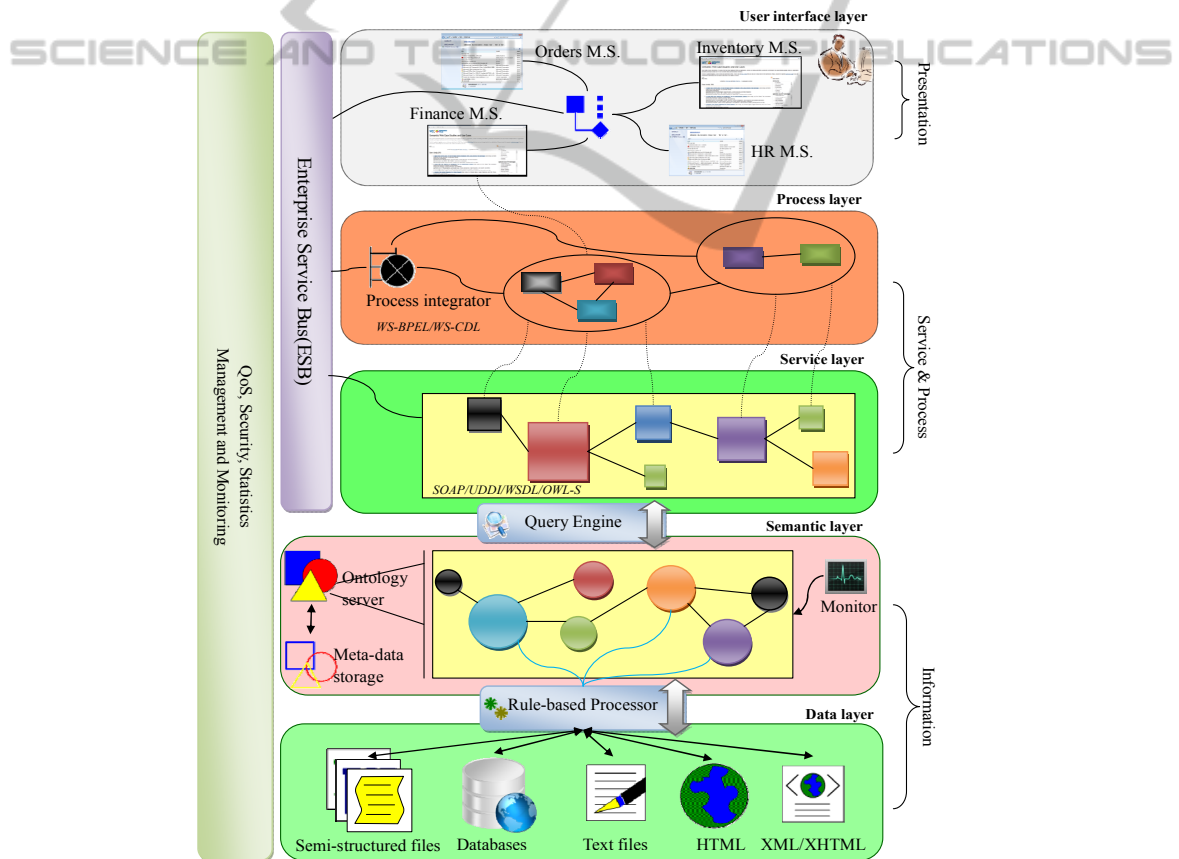


Figure 5: Five layers of the architecture. Another important issue when building semantic is to keep it adaptive and evolutionary. Because the information in data sources changes frequently, such as adding a new database or removing some files. The ontology needs to keep consistent with the real information layer. Even the data itself doesn't change, the meaning of information could change date by date. Because we are dealing with semantics, when the meaning of a word changes, the old ones become meaningless and incorrect, so keeping evolutionary is necessary and important. To do these tasks, a component called monitor is set up to keep the ontology update-to-date and correct.

A meta-data storage is designed to abstract the global ontology vocabularies. Furthermore it could be used to extend the global ontology schema to multiple and hybrid schema when single mode has too much limitations and constraints for a specific future system.

There are many languages for representing ontology, each one has its highlight and constraints, the selection will depend on the specific focus and requirements. (Pulido et al, 2006) and (Maniraj, 2010) give a review of the ontology languages, (Corcho et al, 2002) presents the different ontology languages with detailed comparisons and introduces the tools for building ontologies.

### 3.3.3 Services Layer

Service layer lies above the information and semantics, these services are the existing and new developed ones located in different information systems and enterprises. Traditionally, these services are independent and not related; however, they could be reconstructed to compose into a process that fulfills certain tasks. The upper level will perform the service integration. In this layer, the services are gathered to establish a semantic web services layer, compared with the conventional web services, semantic web services has the ability to understand the services and the relations between them.

A component called query engine is designed for querying and answering between the semantic layer and the services layer. The query engine is ontology oriented, such as, SPARQL (W3C recommendation, 2008), so that it can query the ontology in the semantic layer. An optimizing mechanism is set up in this query engine, for optimizing the query request sent from application level and the answers.

### 3.3.4 Business Process Layer

Business process layer represents the business activities in enterprises, these business processes happen internally or among enterprises. These processes are composed with the atomic services in services layer. Service orchestration and choreography are applied to coordinate and compose services into business process, for example WS-BPEL and WS-CDL are specific description language for business process integration.

WS-CDL is an XML-based language that describes peer-to-peer collaborations of parties by defining, from a global viewpoint, their common and complementary observable behavior; where ordered message exchanges result in accomplishing a common business goal (W3C, 2004). WS-BPEL is

an OASIS standard executable language for specifying actions within business processes with web services, WS-BPEL export and import information by using web service interfaces exclusively (OASIS, 2007).

A business integrator is designed to perform the process integration task, and in charge of connecting to the ESB (Enterprise Service Bus) (Chappell, 2004). The dotted line between the process and the services in the figure 7 is logically connected. It means a certain process is composed with several services, but physically, the process and services are connected via ESB.

### 3.3.5 User Interface Layer

This layer represents the user level, namely, the interface the users interact with, such as the human resources system, orders management system, inventory management orders and etc. Via the defined interface, the applications connect to the ESB and explore the process and service to complete user requirements. New modules and functionalities could be added and developed within the current systems. So this could keep the dependability of the current systems, they don't need to be changed to adapt the new requirements, because the cost for updating the legacy or developed systems is very high, we try to minimize the unnecessary changes and maximize the value-added features.

## 4 DISCUSSION

In this paper, we addressed the importance of semantic and business process for enterprise information system interoperability. Based on the ideology of SOA, we proposed a new architecture to integrate enterprises information systems, which uses semantic web technologies to construct a semantic layer upon data sources, and service orchestration and choreography to integrate business process. The top level is the information systems, with which the end users interact directly. The applications connect to the ESB through the defined interface, and then invoke the processes and services to fulfill business requirements.

Ziegler (2004) sums up the usual methods for data integration by six levels of integration. They are described from the architectural perspective of information systems. In particular, at the user level only "Manual" integration, which has a very low integration degree, is available. The most often used approach is "Uniform Data Access" to solve the data integration problems, which provide user a unified user interface of physically distributed data, such as

portals and mediated query systems. “Common data storage” uses a method to transfer the data sources to a new data storage, warehouse and operational data stores are such kind of instances.

Compared with these integration approaches, interoperability has more the meaning of coexistence, autonomy and federate environment, whereas integration refers to the concepts of coordination, coherence and standardization (Chen, 2008). The interoperable architecture will provide added values for enterprise information systems interoperability. First, due to the semantic technologies used in the architecture, the “understanding” problems can be facilitated among different information systems. Secondly, with SOA, the architecture is loosely coupled, where each layer is only connected through interface and protocols. The data sources, services and business processes could be added or removed without affecting the other parts of the architecture, gaining flexibility for the enterprise. Thirdly, it provides high scalability; this architecture could be applied internally in one enterprise for its sub branches or different departments, as well as among different enterprises and partners. Involved applications and services can be reused ones; also new developed ones can be added easily. We believe this approach is promising regarding web 2.0 orientation of all enterprise applications.

## 5 FUTURE WORK

The future work will include, first, more details to describe the running mechanism of the architecture, such as, the interfaces between each layer. Second, we will elaborate and refine the semantic layer to define how the ontology presents the semantics of information; third, the aspects of enterprises business level (e.g. decision level) will be taken into account to make the architecture adapted the complex situation in enterprises. The architecture will be applied to a project in the following research, to figure out the improvements and get feedbacks from industry.

## REFERENCES

- Bell M. (2008). *Service-Oriented Modeling: Service Analysis, Design, and Architecture*, Wiley & Sons
- Bratt, S. (2007). Semantic Web, and Other Technologies to Watch, [Online], Available: <http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb>, [Dec. 14, 2010]
- Buccella, A., Cechich, A. and Brisaboa, N. (2003) ‘Ontology-based data integration methods: A framework for comparison’.
- Chappell, D. (2004). *Enterprise Service Bus*, O’Reilly
- Chen, D., Daclin, N., (2006) ‘Framework for enterprise interoperability’, *Proc. of IFAC Workshop EI2N 2006, Interoperability for enterprise software and applications* (ed. H. Panetto and N. Boudjlida), Bordeaux, March 21, 2006, pp. 77-88.
- Chen, D., Doumeings, G. and Vernadat, F. (2008) ‘Architectures for enterprise integration and interoperability: Past, present and future’, *Computers in Industry*, vol. 59, no. 7, September , pp. 647-659.
- Corcho O., Fernández-López M., Gómez-Pérez A. (2003). ‘Methodologies, tools and languages for building ontologies. Where is their meeting point?’, *Data & Knowledge Engineering*, vol. 46, no. 1, pp. 41-64.
- Gruber, T. (1993). ‘A Translation Approach to Portable Ontology Specifications’, *Knowledge Acquisition*. Available: <http://tomgruber.org/writing/ontolingua-kaj-1993.pdf>.
- Izza, S., Vincent, L. and Burlat, P. (2008), ‘Exploiting semantic web services in achieving flexible application integration in the microelectronics field’, *Computers in industry*, vol.59 , no. 7, pp. 722-74
- Maniraj V., Sivakumar R. (2010). ‘Ontology Languages – A Review,’ *International Journal of Computer Theory and Engineering*, vol. 2, no. 6, pp887-891.
- Peltz, C. (2003), ‘Web service orchestration and choreography – a look at WSCI and BPEL4WS’, *Web Service Journal*. Available: <http://wpage.unina.it/rcanonic/didattica/at/documenti/wsOrchestration.pdf>
- Pulidoa J.R.G., Ruizb M.A.G., Herrerac M.A.G., Cabellod E., Legrande S., Ellimanf D. (2006). ‘Ontology languages for the semantic web: A never completely updated review’, *Knowledge-Based Systems*, vol. 19, no. 7, pp. 489-497.
- W3C recommendation, Query Language for RDF (2008), [Online], Available: <http://www.w3.org/TR/rdf-sparql-query/> [Dec. 20, 2010]
- W3C study cases (2010) ,[Online], Available: <http://www.w3.org/2001/sw/sweco/public/UseCases/> [Dec. 14, 2010]
- Wache, H., Gogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. and Hubner, S. (2001). ‘Ontology-based integration of information – a survey of existing approaches’, *Wache2001* , pp. 108-117.
- Web Services Business Process Execution Language (WS-BPEL) OASIS Standard (2007), [Online], Available: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html> [Apr.11, 2007]
- Zacharewicz, G., Labarthe, O., Chen, D. and Vallespir B. (2009). ‘A Multi Agent/HLA Platform for Enterprises Interoperability: Short-Lived Ontology Based’ in *Electronic Supply Network Coordination in Intelligent and Dynamic Environment: Modeling and Implementation*, IGI Global
- Ziegler, P. and Dittrich, K. (2004) ‘Three decades of data integration – all problems solved?’, In *18th IFIP World Computer Congress*, Volume 12, *Building the Information Building the Information Society*, vol. 2004, pp. 3-12.