

A Unified Framework for Enterprise Integration

An Ontology-Driven Service-Oriented Approach

Saïd Izza, Lucien Vincent, Patrick Burlat
Laboratoire G2I, Ecole des Mines de Saint-Etienne
158 Cours Fauriel, Saint-Etienne, France
izza@emse.fr, vincent@emse.fr, burlat@emse.fr

Abstract. Today, enterprise integration (EI) is still facing many problems and the crucial one is the semantic heterogeneity. This problem is not correctly addressed by today's EI solutions that focus mainly on the technical and syntactical integration. Addressing the semantic aspect will promote EI by providing it more consistency and robustness. Some efforts are suggested to solve the semantic problem at different level, but they are still not mature. This article will propose an approach that combines both ontologies and web services in order to address this hard integration problem.

1 Introduction

Over the last decade, a new technology and a new industry sector, typically known as enterprise application integration (EAI) technology, have emerged as a sub-field of enterprise integration (EI) [22]. In essence, EAI technologies provide tools to interconnect multiple and heterogeneous enterprise information systems (EIS) such as ERP (Enterprise Resource Planning) systems, CRM (Customer Relationship Management) systems, SCM (Supply Chain Management) systems, and legacy systems. The most difficulty of this interconnection is due to the fact that the integrated systems were never designed to work together.

More recently, Web Services (WS) have emerged with the advent and the evolution of the Internet and they provide a set of standards for EAI. Even if WSs are not fully mature, they seem to become the *linga franca* of EAI [14]. This will notably make integration simpler and easier through using web protocols and standards.

Despite the whole range of available tools and widespread standards adoption, the main goal of EAI which is the semantically correct integration of EISs is not yet achieved [6]. Indeed, EAI still provides technical and syntactical solutions but does not address correctly the semantic problem which is the hard problem in EI.

This paper will propose an approach in order to address the semantic integration in the context of EAI. In the rest of this paper, we will firstly present the integration problem; secondly, we will briefly review the current state-of-the-art in EAI through presenting two kinds of solutions: traditional EAI systems and Web-Services-based EAI systems. Then, we will describe some aspects of our work attempting to provide a

solution for the integration problem, and finally before concluding, we will outline some important related works.

2 Integration Problem

Enterprise information system (EIS) is defined as an enterprise application system or an enterprise data source that provides the information infrastructure for an enterprise. Typically, an EIS can take many different types including batch applications, traditional applications, client/server applications, web applications, application packages, relational databases and so on [17]. These systems are often materialized in enterprise reality in form of relational databases, ERP (Enterprise Resource Planning) systems, CRM (Customer Relationship Management) systems, SCM (Supply Chain Management) systems and legacy systems.

An appropriate characterization of EISs in context of EAI is that these systems are HAD (heterogeneous, autonomous, distributed) [16] [6]:

- *Heterogeneous* means that each EIS implements its own data and process model.
- *Autonomous* refers to the fact that EISs run independently of any other EIS.
- *Distributed* means that EISs locally implement their data model which they generally do not share with other EISs.

The consequence of the characteristics above is that EISs are standalone software entities which constitute what [31] labels "islands of information" and what [24] calls "islands of automation". Of course, none of the islands contain enterprise-wide processing and information, which makes it then necessary to integrate the islands in order to create unified enterprise information and process system. In this case, any form of integration of these islands (EISs) must happen outside of the involved islands (EISs), by using integration systems such as EAI systems. This integration consists then in interconnecting the interfaces of each EIS using technologies supported by the integration systems such as queuing systems, file systems, databases or remote invocations.

Despite the importance of all the problems described above, we focus only on the heterogeneity problem, precisely the semantic heterogeneity problem which is the hard problem of enterprise integration in general and EAI in particular [6].

3 Today's EAI Solutions

The fundamental EAI drivers are related to the existence of EISs which are HAD (heterogeneous, autonomous and distributed) systems. Generally, this is the consequence of the emergence of e-business, enterprise mergers and consolidations, and the rise of application packages and COTS (Commercial On The Shelf) [11]. In the rest of this paper, we consider only two main solutions which are the most important ones in the context of EAI: traditional EAI systems and WSs. These solutions can fulfill major integration requirements such as data synchronization and integration, business process execution, fast deployment of new applications and so on.

3.1 Traditional EAI Systems

Currently, EAI systems are based on a lot of technologies such as: message brokers, process brokers, message-oriented middleware (MOM), etc. [17]. Even if EAI systems may differ from a technological point of view, the main functionalities remain the same, and we can mainly distinguish five components which constitutes a hierarchy of layers and provide respectively transport services, connectivity services, transformation services, distribution services and process management services. The transport services provides communication of data, and it is generally based on asynchronous middleware technology such as message queuing systems. The connectivity services provides adapters/connectors that connect EISs to the transport services layer. The transformation services provide translation of the exchanged data between EISs in order to conciliate the syntactical differences. The distribution services is responsible for routing the messages. At last, process management services provides mechanisms to manage and coordinate business processes by using workflow or BPM (Business Process Management) engines.

The principle of EAI systems is based on using interfaces (adapters/connectors) to integrate EISs. The interfaces convert all traffic to canonical formats and protocols. These interfaces constitute the only mean to access EISs and they can occur in different levels: user-interface level, business logic level and data level [21].

Although EAI systems address technical and syntactical integration, nevertheless they must address the semantic level which is more difficult and which can provide more added-value. Since the EIS's interfaces are generally immutable, both from syntactic and semantic perspective, the EAI systems must be able to conciliate between messages of the different EISs on a syntactic level as well as on the semantic level. On the syntactic level, EAI systems must be able to receive any particular format from any EIS, transform it and transmit it to any other EIS. On the semantic level, the same process must happen with the help of semantic mediation in order to ensure that the semantics of messages will be preserved during exchanges.

Today, no traditional EAI system can provide mechanism that correctly supports semantics. In most cases, data is passed between EISs by-value, and in general no shared semantic concepts are explicitly used to define semantics through different messages or to describe the functions that are provided. As far as we know, only few systems such as Unicorn system [35] can support semantic concept with the help of a canonical and corporate ontology. Furthermore, constructing such an ontology is costly if not impractical especially when the enterprises are large and the EISs usually change and evolve over time. So this approach do not really solve the integration problem.

3.2 Web Services

WSs are considered as a result of convergence of Web with distributed object technologies [2]. They are defined as an application providing data and services to other applications through the Internet [18]. WSs promotes an SOA (Service-Oriented Architecture) that distinguishes fundamentally three roles: service provider, service

requestor and service broker; and three basic operations: publish, find and bind, and any particular EIS can play any or all these roles [20].

WSs constitute the most important instantiation of the SOA model. They can be deployed inside (in the context of EAI) or outside (in the context of B2B) the enterprise. In all cases, WSs are published with appropriate URLs by WS providers over the internet or intranet. Once published, these WSs are accessible by WS consumers via standards Web such as HTTP (HyperText Transfer Protocol), SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) and UDDI (Universal Description, Discovery and Integration) [3]. In addition to this, WSs can be used for integrating EISs via standards such as BPEL (Business Process Execution Level) or WSFL (Web Service Flow Language).

WSs are increasingly playing an important role in EAI [20] [30]. They are precisely taking the role of the EAI bus to which all EISs connect [3]. In this case, the connecting EISs are often wrapped by WSDL interfaces, or they are in some cases converted to modern web-service-based-applications via a migration process. These wrappers and migrations are generally done by existing web service toolkits.

WSs are very promising in solving the integration problem [23]. Today, some new integration products based on WSs standards exist and will certainly replace in the near future the proprietary solutions that are the traditional EAI systems. Examples of such new products are Cape Clear, XML Global, Colloxa, etc. [4]. A widespread adoption of WSs may enhance the evolving technology standards.

Even if WSs are promising, however they do not correctly address the semantic definition that is currently somewhat supported mainly by UDDI registries with the help of some standard taxonomies such as NAICS (North American Industrial Classification Scheme), UN/SPSC (Universal Standard Products and Services Classification) and ISO 3166 (Geographical taxonomy) [9]. In addition, WSs do not provide neither data nor behavior mediation [8] [12] [7]. These drawbacks are mainly due to the lack of service ontology and mediation support in current WSs. This lack penalizes the efficiency of current WS integration in the context of EAI.

Nevertheless, we must mention the existence of some important initiatives around the concept of semantic web service [33] [9] [25] that aims to bridge the above current WS gap such as OWL-S [37], BPEL [10], WSMF [12], METEOR-S [27]. But, most of these efforts do not provide mature concepts for both data and behavior mediation in the particular context of EAI. The approach we propose in the following section takes and improves some of the best ideas emerged from these efforts and which we attempt to apply to the EAI context.

4 ODSOI Approach

This section succinctly describes some important characteristics of our approach called ODSOI (Ontology-Driven Service-Oriented Integration) that rely on the use of both ontologies and WSs technologies, and that aims to extend the state-of-the-art in EAI in order to address the semantic problem.

4.1 General principles

First of all, ODSOI approach is a solution to the information system integration problem. This means that our approach addresses the heterogeneity problem by providing an *ontology-driven* integration solution that is based on ontology mediation. Indeed, our approach is *service-oriented* (based on SOA) since it uses WSs for integrating EISs. The result architecture integration that we suggest is called ODSOA (Ontology-Driven Service-Oriented Architecture). This latter extends SOA with a semantic layer that aims to enhance service mediation in the context of EAI.

Furthermore, our approach can support both *static* and *dynamic oriented integration*. This latter means that the binding services of target EISs can be performed at run time according to special binding rules, contrarily to a static integration solution where an integration scenario predefines every detail of potentially connectable EISs (Kwak, 2002). The reasons of this choice are that the dynamic solution is sometimes more flexible and scalable form of integration than static one. In addition to this, the integration scenario becomes simpler and the configuration EIS interfaces can be changed and new systems can be added easily, even while the EAI system is running.

4.2 Global Architecture

The ODSOA concept provides a unified framework in order to integrate EISs. In this framework, three main types of services are defined: data-services, functional-services and business-services. This different types can respectively address data, application and process integration.

Data-Services (DS) are services that expose data sources (EDS – Enterprise Data Sources) as services. *Functional-Services* (FS) are services that expose application systems, fundamentally functional systems (EAS – Enterprise Application Systems) as services. *Business-Services* (BS) are defined as the combination of the above services in order to expose business processes (EPS – Enterprise Process Systems) as services. Our service typology enrich somewhat the topology proposed in [34] and which is based on two types that are SaaS (Software-as-a-Service) and DaaS (Data-as-a-Service).

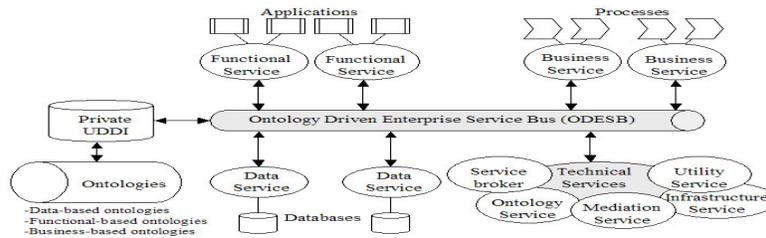


Fig. 1. Global view of ODSOA Architecture

Figure 1 which is a particular SOA recapitulates these important types of services. Indeed, there are, of course, some other important technical services that are mainly

service broker, ontology services, mediation services, infrastructure services and utility services. Some of them will be described below.

A cross section of the integration bus (also called ODESB – Ontology-Driven Enterprise Service Bus) (figure 2) shows many concentric standard layers such as *Transport layer*, *Exchange layer*, *Description layer*, *Registry layer* and *Transversal layer*. In addition to these standard and currently existing layers, we suggest to adopt, in a similar way as semantic web service concept [33], an other layer, called *semantic layer* and which contains two sub-layers that are *ontological layer* and *mediation layer*.

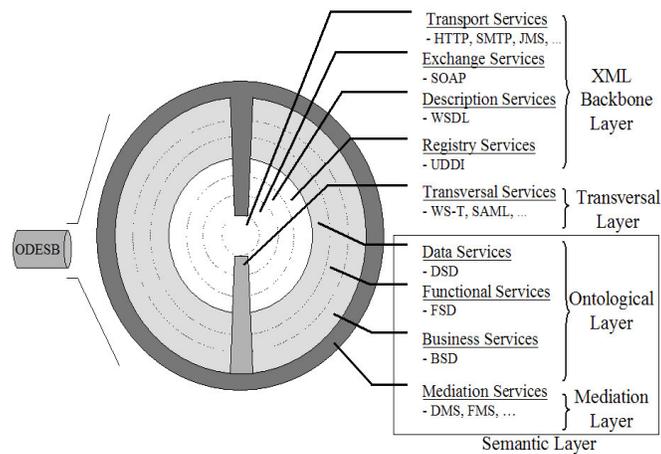


Fig. 2. Cross Section of the ODESB Bus

The Ontological layer aims to describe the three fundamental types of services described above using specific descriptions such as DSD (Data Service Description) for DSs, FSD (Functional Service Description) for FSs, and BSD (Business Service Description) for BSs. All these descriptions exploit some ontologies and are specializations of OWL-S (Web Ontology Languages-Services) that constitutes a semantic orientation to the description of WSs, contrary to WSDL (Web Service Description Language) which provides only a description based on a syntactic orientation.

The mediation layer provides mechanisms to perform the resolution of semantic differences (semantic mediation). In the next sub-section, the services of semantic layer which are the important ones in our approach will be developed.

4.3 Semantic Layer Services

Semantic services are the main services that address the semantic problem. They are divided into ontology services and mediation services. Each type of services will be described below.

4.3.1 Ontology Services

The principle of ODSOA is based on the semantic services that perform mediation processes in order to resolve mainly the semantic heterogeneity problem. These processes exploit some formal ontologies which take important part in our architecture. Ontology services aims to define the semantic description of services using ontology concepts. According to Gruber, an ontology is defined as an explicit and formal specification of a conceptualization [15]. In the context of information integration, two fundamental approaches can be selected to achieve semantic integration: shared ontology approach and non-shared (local) ontology approach [38]. In general, we use an hybrid approach that combines the two approaches and that consists to map local ontologies onto a shared ontology [38].

For our purpose, we have chosen a variant of the hybrid approach. This can be motivated by the fact that in a large enterprise of autonomous EISs, the definition of a single set of ontology standards that are suitable for everyone is nearly impossible. Indeed, this can also be motivated by the difficult leveraging when adopting multiple independent ontologies.

In our approach, we have defined three major types of ontologies that are similar to those briefly cited in [21]: information or data-based ontologies, behavior or functional-based ontologies and process or business-based ontologies (figure 3). These ontologies are used in service descriptions such as DSD (Data Service Description), FSD (Function Service Description) and BSD (Business Service Description) to explicit the semantics of respectively DSs, FSs and BSs.

Data-based ontologies are the most basic ones. They provide semantic description of the data (DSs). These ontologies are required in all cases, no matter if we leverage functional-based or business-based ontologies. This means that, data integration is a precondition for any other integration type.

Functional-based ontologies define semantic description around functions that provided by the multiple EISs (FSs) and that can be remotely invoked. These ontologies are generally required in order to provide a better reuse of functionalities.

Business-based ontologies define semantic description around coordinating business processes. These ontologies are generally required in order to integrate business processes.

In addition to this, ontology services are based on the context of a service (service-context) which is defined by a set of ontologies related to the concerned service. This service-context is also called local ontology which means that there are several ontology levels. This is generally appropriate within a large and complex enterprise with several different business domains such as the studied society that is large and concerns the multidisciplinary microelectronics area. For our purpose, three ontology levels have been identified: *local level*, *domain level* and *global level* (figure 9).

In essence, local ontologies concerns services, whereas domain ontologies concern the generalization of local ones that belong to the same domain (Production, Metrolology, Packaging, etc.) and they can serve in aligning some local ontologies. At last, global ontology is considered as generalization of domain ontologies. It is the root of the ontology hierarchy, and it can serve in aligning domain ontologies and may also be used in B2B integration that can constitute a natural extension of our approach.

The ontology taxonomy defines somewhat a clusterization process which is firstly used in [36]. This clusterization is an important concept which defines the ontology structuration within an enterprise and can be used in order to master the ontology construction and evolution. We call this clusterization: *ontology urbanization* and it takes an important role in our integration approach.

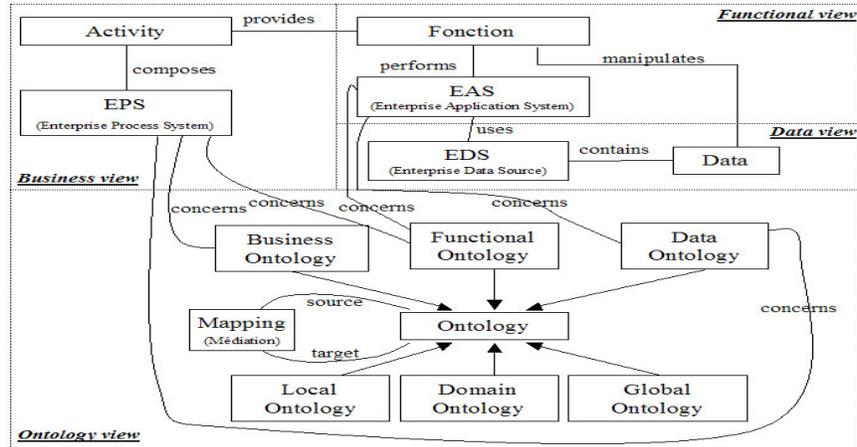


Fig. 3. ODSOI ontology taxonomy

4.3.2 Mediation Services

Mediation services are generally invoked by service brokers in order to perform matching or resolution of semantic heterogeneity between services. They exploit the descriptions provided by the ontology services described above.

Since we use an hybrid ontology approach, this require the integration and mediation of ontologies which are performed by Ontology Mediation Services (OMS) and that are based on ontology mapping [19]. This latter is the process whereby two or more ontologies are semantically related at conceptual level. According to the semantic relations defined in the mappings, source ontology instances can then be transformed (or matched with) into target ones (Noy, 2004).

In addition to OMS and according to the above different fundamental types of services, we can distinguish mainly three other types of mediation services: Data Mediation Service (DMS), Functional Mediation Service (FMS), Business Mediation Service (BMS). These mediation services aims to mediate respectively between DSs, FSs, BSs and they are based on OMS that match and mediate between different ontologies. To be performed, OMS can exploit two particular utility services that are inference service and matching service. These particular services can be respectively supported by academic or commercial inference engine and matching tool. For the initial prototype that is ongoing, we decide to use Racer engine [29] and OLA (OWL Lite Alignment) matcher [28] that seems be appropriate to our approach.

4.4 Overview of the Integration Process

ODSOI approach provides a uniform framework to integrate EISs at different levels: data, application and process integration level that are provided by three specific services or sub-frameworks that are respectively ODSODI (Ontology-Driven Service-Oriented Data Integration) framework, ODSOAI (Ontology-Driven Service-Oriented Application Integration) framework and ODSOBI (Ontology-Driven Service-Oriented Business Integration) framework. Figure 4 illustrates the general principle of the three components of the global framework (ODSOI framework).

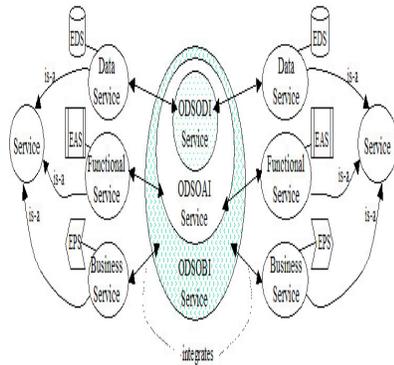


Fig. 4. General vision of ODSOI framework

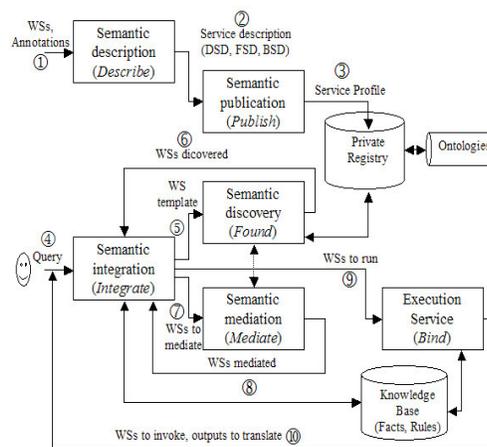


Fig. 5. Generic integration process

DSs can be integrated with the ODSODI framework by using data ontology mediation, whereas FSSs can be integrated in similar way by using both data and functional ontology mediation. At last, in case of process integration, BSs can be integrated by using data, functional and business ontology mediation. These three sub-frameworks define a stratification so that each level include its inferior levels. Since we can not detail the different sub-frameworks in this paper, so we just give a generic synopsis about the integration process.

The heart of integration process is realized by mediation services with the help of a service broker. This latter is composed of different technical services such as integration service, discovery service, and execution service. The generic scenario of the integration process (figure 5) starts once the Ws have been described (semantic description) by using ontology services. After that, they are published (semantic publication) in enhanced private UDDI registries (by ontology services) and then they can be discovered (semantic discovery) by discovery service in order to carry out the realization of a given task modeled as a service template (that corresponds to a user or service query) by the integration service. The discovery service can use mediation service in order to perform the semantic matching. The invoked mediation services exploit a similarity function that calculate rapprochement between ontology concepts and then between the involved characteristics of services. Once the desired Ws have been

discovered, they are also mediated in order to resolve the semantic heterogeneity differences by the mediation service. Finally, the mediated services are executed by the execution service and can invoke the integration service which can then perform another similar loop of the integration process. Furthermore, the above process include dynamic integration which is made possible by some binding rules contained in the knowledge base. These rules exploits mainly the characteristics of services such as preconditions, effects and so on.

5 Related works

The integration problem constitutes a hard and complex domain addressed by several communities such as database community, software engineering community, data mining community, enterprise information system community and so on. In this section, we will briefly and without any exhaustiveness present some important related works.

First, in the context of data integration, there are many general works which use ontology-based approaches such as COIN (Context Interchange) project [13], OBSERVER (Ontology Based System Enhanced with Relationships for Vocabulary Heterogeneity Resolution) project [26], INFOSLEUTH project [39], BUSTER project [32] and so on. All these work are not concerned about the mediation in the context of SOA.

In addition to the listed related works above, there are some other works that are addressing the WS viewpoint such as Active XML from GEMO project [1] and SODIA (Service-Oriented Data Integration Architecture) from IBHIS (Integration Broker for Heterogeneous Information Sources) project [34]. Active XML extends XML language by allowing embedding of calls to WSs. SODIA is an implementation of Federated Database System in the context of WSs. These work do not support any mediation services.

Furthermore, in the context of application and process integration, some important initiatives and works exist such as OWL-S [37], BPEL [10], WSMF [12], METEOR-S [27]. OWL-S provides an ontology markup language in order to semantically describe capabilities and proprieties of WSs. BPEL is a standard providing a language to define business processes that can be used in application integration. WSMF, METEOR-S are initiatives that provide frameworks in order to support the concept of semantic web service which are defined WSs with a formal description (semantics) that can enable a better discovery, selection, composition, monitoring, and interoperability [8]. WSMF are based on the extension of DAML-S and OWL-S whereas METEOR-S is an effort which provides semantic web services through the extension of existing standards such as WSDL and UDDI. But, most of these efforts do not provide mature concepts for mediation in the context of EAI.

Finally, none of the related works provide a unified framework in the context of SOA to integrate semantically both data, application and processes.

6 Conclusion

The semantic integration of enterprise information systems is a crucial problem that can concern data, applications and processes. This problem needs semantic mediation and is best resolved in the context of service oriented architectures.

This paper has focused on proposing a unified approach for enterprise application integration that exploit both ontology mediation and web services. This approach called ODSOI (Ontology-Driven Service-Oriented Integration) aims to extend the current web services stack technology by a semantic layer offering semantic services that can define the service semantics and also perform semantic mediation in the context of EAI. A typology of services and also of ontologies have been suggested, and a global vision of the integration framework is described. We intend to materialize in the future our work with an initial prototype addressing the data aggregation problem within the study society (*International Microelectronics Society*) which provides us an operational environment in order to verify the pertinence and validity of our approach.

References

1. Abiteboul S., Benjelloum O., Milo T., 2002. Web Services and Data Integration. *INRIA report*.
2. Acharya R., 2003. EAI: A business perspective, *eAI Journal*. [Online]. Available at: www.bijonline.com/PDF/Apr03Acharya.pdf.
3. Amjad U., 2004. The Emerging Role of the Web for Enterprise, *Proceedings of the IEEE*, 92(9), pp. 1420-1438..
4. BIJOnline, www.bijonline.com.
5. BPEL4WS, <http://www-128.ibm.com/developerworks/library/ws-bpel/>.
6. Bussler C., 2003. The Role of Semantic Web Technology in EAI. *In the Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*.
7. Bussler C., 2004. Semantic Web Services. *ICWE'04*. Munich.
8. Cardoso J., Bussler C., Sheth A., Fensel D., 2002. Semantic Web Services and Processes: semantic Composition and Quality of Service. *Tutorial at Federated Conferences*.
9. Dogac A., 2004. Semantic Web Services, *ICWE'04*, Munich.
10. EBPML, www.ebpml.org.
11. Erasala N., Yen D. C., Rajkumar T. M., 2003. Enterprise Application Integration in the electronic commerce world. *Computer Standards & Interfaces* 25 (69-82). Elsevier.
12. Fensel D., Bussler C., 2002. The Web Service Modeling Framework WSMF. *In Electronic Commerce Research and Applications*, 1(2). Elsevier, Sciences B. V.
13. Goh C. H., 1997. Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources. *Phd Thesis, MIT*.
14. Gorton I., Thrman D., Thomson J., 2003. Next Generation Application Integration: Challenges and New Approaches, *Information Sciences and Engineering, Richland*.
15. Gruber R. T., 1993. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *KSL, Stanford University*.

16. Hasselbring W., 2000. Information System Integration, *Communications of the ACM Vol 43(6)*, 33-38.
17. Izza S., Vincent L., Burlat P., Solignac H., Lebrun P. Intégration d'applications: Etat de l'art et Perspectives. *2èmes Journées d'Informatique pour l'Entreprise JIE 2' 04 - Algérie*.
18. Kadima H., 2003. Monfort V., Les Web Services. *Dunod*.
19. Kalfoglou Y., Schorlemmer M., 2003. Ontology Mapping: The State of The Art. *The Knowledge Engineering Review, Vol. 18(1)*, 1-31. Cambridge University Press.
20. Kontogiannis K., Smith D., O'Brien L., 2002. On the Role of Services in Enterprise Application Integration. *In the Proceedings of the 10th International Workshop on Software Technology and Engineering*.
21. Linthicum D. S., 2004. Next Generation Application Integration. *Addison-Wesley*.
22. Lithicum D. S., 1999. Enterprise Application Integration, *Addison-Wesley*.
23. Lublinsky B., Tyomkin D., 2003. Dissecting Service Oriented Architectures. *BI Journal*.
24. Lublinsky B., 2001. Achieving the Ultimate EAI Implementation. *eAI Journal*.
25. McIlraith S., Son T., Zeng H., 2001. Semantic Web Services, *IEEE Intelligent Systems*, pp 46-53.
26. Mena E., Kashyap V., Sheth A., Illarramendi A., 1996. OBSERVER: An approach for query processing in global information systems based on interoperability across pre-existing ontologies. *In Proceedings 1st IFCIS International Conference on Cooperative Information Systems, Brussels*.
27. METEOR-S, <http://lsdis.cs.uga.edu/Projects/METEOR-S>.
28. Noy N. F., 2004. Semantic Integration: A Survey Of Ontology- Based Approaches. *Stanford Medical Informatics. Stanford*.
29. OLA, www.iro.umontreal.ca/~owlola/alignment.html.
30. Racer, <http://www.racer-systems.info>.
31. Roszko A., Kontogiannis K., 2003. Revolutionizing EAI with Enterprise Web Services. *STEP'03, IEEE International Conference on Software Maintenance, ICSM'03*.
32. Stonebraker M., 1999. Integrating Islands of Information, *eAI Journal*.
33. Stuckenschmidt H., Wache H., Vögele T., Visser U., 2000. Enabling technologies for interoperability. *Workshop on the 14th ISCSEP, pages 35-46, Germany*.
34. SWSI. Semantic Web Services Initiative, www.swsi.org.
35. Turner M., Zhu F., Kotsiopoulos I., Russel M., Bennett K., Brereton P., Keane J., Rigby M., 2004. Using Web Service Technologies to Create an Information Broker: An Experience Report. *26th International Conference on Software Engineering (ICSE'04). IEEE Computer Society. pp 552-561*.
36. Unicorn, www.unicorn.com.
37. Visser P., Tamma V., 1999. An experience with ontology clustering for information integration. *In Proceedings of the IJCAI-99 Workshop on Intelligent Information Integration, Sweden*.
38. W3C, www.w3.org.
39. Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner S., 2001. Ontology-Based Integration of Information - A Survey of Existing Approaches. *Intelligent Systems Group, Germany*.
40. Woelk D., Tomlinson C., 1994. The infosleuth project: intelligent search management via semantic agents. *In Second World Wide Web Conference '94: Mosaic and the Web*.