A UNIFIED FRAMEWORK FOR APPLICATION 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

Keywords: Enterprise application, integration, web service, ontology, mediation.

Abstract: The crucial problem of Enterprise Application Integration (EAI) is the semantic integration problem. This latter is not correctly addressed by today's EAI solutions that focus mainly on the technical and syntactical integration. Addressing the semantic aspect will promote EAI by providing it more consistency and robustness. Some efforts are suggested to solve the semantic problem, but they are still not mature. This article will propose an approach that combines both ontologies and web services in order to overcome some issues related to the semantic integration problem.

1 INTRODUCTION

Over the last decade, a new technology typically known as Enterprise Application Integration (EAI), have emerged. In essence, EAI technologies provide tools to interconnect multiple and heterogeneous enterprise application systems (EAS) such as ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), SCM (Supply Chain Management), and legacy systems. The most difficulty of these interconnections is 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. 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 EASs, is not yet achieved. Indeed, EAI still provides technical and syntactical solutions but does not address correctly the semantic problem, which constitutes the real integration problem.

Semantic integration becomes very important in order to overcame semantic heterogeneities within EAI, and which mainly concern both data and behavior of EASs. Although there is some related works, which concern semantic integration, but there has been no mature solution that deals correctly with integration problem.

In this paper, we will focus on the semantic problem in the context of EAI. Our approach is based on an extension of service-oriented architecture (SOA), called ODSOI (Ontology-Driven Service-Oriented Integration) and which is mainly based on WSs and ontologies. The rest of this paper is organized as follows. Firstly, we will present the integration problem. Secondly, we will briefly review the current state-of-the-art in EAI through presenting two major kinds of solutions: traditional and Web-Services-based EAI systems. Finally and before concluding, we will describe some aspects of our work which attempts to provide a solution for the integration problem.

2 THE INTEGRATION PROBLEM

Enterprise application systems (EAS) can take many different types including batch applications, traditional applications, client/server applications, web applications, application packages (Izza, 2004). These systems are often materialized in enterprise reality in form of ERP, CRM, SCM, and legacy systems.

An appropriate characterization of EASs in the context of EAI is that EASs are HAD (heterogeneous, autonomous and distributed) systems (Bussler, 2003):

- Heterogeneous systems mean that each EAS implements its own data and process model.
- Autonomous systems refer to the fact that each

Izza S., Vincent L. and Burlat P. (2005).
A UNIFIED FRAMEWORK FOR APPLICATION INTEGRATION - An Ontology-Driven Service-Oriented Approach.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 165-170
DOI: 10.5220/0002521601650170

EAS runs independently of any other EAS.

- Distributed systems mean that each EAS locally implements its data model, which it generally do not share with other EAS.

The consequence of these characteristics is that EASs are generally standalone software entities, which form what we often call islands of information and automation. In this case, any form of integration of the EASs must happen outside of the involved EASs, by using integration systems such as EAI systems. This integration consists in interconnecting the interfaces of each EAS using technologies supported by the integration systems such as queuing systems, databases or remote invocations.

The characteristics of EASs form the main reasons of the existence of the integration problem, and the more these characteristics are extremes, the more the integration become hard and complex. Despite the importance of the problems described above, we will focus, in this paper, only on the heterogeneity problem, precisely the semantic heterogeneity problem, which is the hard problem of enterprise integration in general, and EAI in particular (Bussler, 2003).

Since EASs are HAD, a semantic mediation is needed in order to achieve their integration. Its aim is to resolve all the semantic conflicts that can arise between the exchanged data, and also between invoked behavior interfaces. Indeed, data semantic mediation provides mechanisms to preserve the meaning of the data during the flow exchanges between EASs (adressing), whereas behavior semantic integration provides mechanisms to resolve the semantic behavior interface heterogeneity when EASs invoke each other.

Furthermore, the integration problem is more complicated by our industrial context concerned by a complex enterprise in the multidisciplinary microelectronics area. This particular context is mainly characterized by several and heterogeneous knowledge domains that needs sophisticated semantic mediation in order to achieve the integration process.

3 INTEGRATION SOLUTIONS

In this section, we will describe the major existing EAI solutions, which will be followed by some pertinent related works about EAI.

3.1 Today's EAI Solutions

In this paper, we will consider only two main important solutions in the context of EAI: traditional EAI systems and WSs. These solutions can fulfill major integration requirements such as data synchronization, business process execution, reconciliation of technical and syntactic differences, fast deployment of new applications and so on.

3.1.1 Traditional EAI Systems

Currently, EAI systems are based on a lot of technologies such as: message brokers, process brokers, message-oriented middleware, etc. 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 provide respectively transport services, connectivity services, transformation services, distribution services and process management services (Erasala, 2003).

The principle of EAI systems is based on using interfaces (connectors) to integrate EASs. The interfaces convert all traffic to canonical formats and protocols. These interfaces constitute the only mean to access EASs, and they can occur in different levels: user-interface level, business logic level and data level (Linthicum, 1999).

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. Today, no traditional EAI system can provide mechanism that correctly supports semantics. In best cases, data is passed between EASs by-value, and in general no shared semantic concepts are explicitly used to define semantics through different messages or to semantically describe the behavior that is provided.

3.1.2 Web Services

WSs are considered as a result of convergence of Web with distributed object technologies. They are defined as an application providing data and services to other applications through the Internet (Kadima, 2003). WSs promote an SOA (Service-Oriented Architecture) that is based fundamentally on three roles: *service provider*, *service requestor* and *service broker*; and three basic operations: *publish*, *find* and *bind*, and any particular EAS can play any or all these roles (Kontogiannis, 2002).

WSs constitute the most important concretization of the SOA model. They can be deployed inside (EAI) or outside (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, SOAP, WSDL and UDDI. In addition to this, WSs can be used for integrating EASs via standards such as BPEL or WSFL.

WSs are very promising in solving the integration problem. 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 (BIJOnline, 2004).

Even if WSs are promising, they do not correctly address the semantic aspect that is currently somewhat supported by UDDI registries with the help of some standard taxonomies such as NAICS, UN/SPSC and ISO 3166 (Dogac, 2004). In addition, WSs do not provide neither data nor behavior mediation (Cordoso, 2002) (Fensel, 2002) (Bussler, 2004). These drawbacks are due mainly 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.

3.2 Related works

Recently, the importance of WSs has been recognized and widely accepted both by industry and academic research. This section will review some important related works about enterprise integration, mainly those that concerns WS-based integration and ontology-based integration.

In the context of data integration, there are many general works which use ontology-based approaches such as COIN project (Goh, 1994), OBSERVER project (Mena, 1996), INFOSLEUTH project (Woelk, 1994), BUSTER project (Stuckenschmidt, 2000) and so on. These works are ontology based but they 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 (Abiteboul, 2002) and SODIA from IBHIS project (Turner, 2004). 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 works do not support any mediation services.

In the context of application and process integration, some important initiatives and works exist around the semantic web service concept (Dogac, 2004) (McIlraith, 2001) that aim to bridge the current WS gap such as OWL-S (W3C, 2004), BPEL (EBPML, 2004), WSMF (Fensel, 2002], SWSI (SWSI, 2004), METEOR-S (METEOR-S, 2004). 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 and SWSI are initiatives that provide frameworks in order to support the concept of semantic web service. METEOR-S is an effort, which provide semantic web services through the extension of WS standards (WSDL, UDDI). But, most of these efforts do not provide mature concepts for mediation, particularly in the context of EAI.

4 ODSOI APPROACH

This section will succinctly describe the important characteristics of our approach called ODSOI that aims to extend the state-of-the-art in EAI in order to address the semantic problem.

4.1 Global Architecture

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 a mediation-based solution using ontology concept. Indeed, our approach is based on service-oriented since it uses WSs for integrating EASs. The architecture integration that we suggest is called ODSOA (ODSO Architecture). This latter extends SOA with a semantic layer that aims to enhance service mediation in the context of EAI.

The ODSOA concept provides a unified framework in order to integrate EASs. In this framework, three main types of services (*Fundamental-Services*) are defined: *Data-Services*, *Functional-Services* and *Business-Services*. This different types can respectively address data, application and process integration.

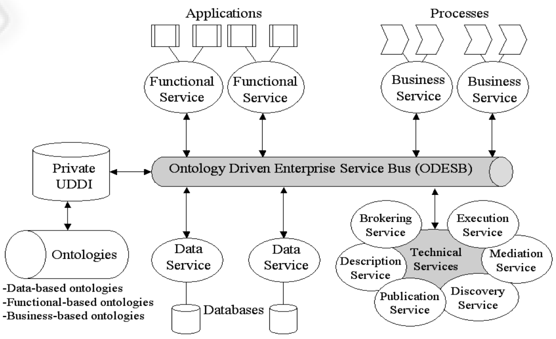

Figure 1: Global View of ODSOA Architecture.

Data-Services (DS) are services that expose data sources as a service. *Functional-Services* (FS) are services that expose application systems, fundamentally functional systems (software that can perform enterprise functions such as administrative and technical ones). *Business-Services* (BS) are defined as the combination of the above services in order to expose business processes. Our service typology can be seen as an extension of the one proposed by (Turner, 2004) which distinguishes two concepts:

SaaS (Software-as-a-Service) and DaaS (Data-as-a-Service).

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 *Brokering-Services*, *Description-Services*, *Mediation-Services*, *Publication-Services*, *Discovery-Services* and *Execution-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 existing standard layers such as *Transport layer, Exchange layer, Registry layer* and *Transversal layer*.

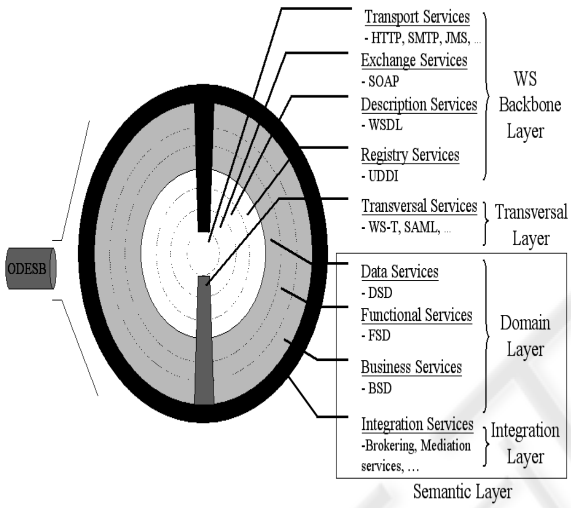

Figure 2: Cross Section of the ODESB Bus.

In addition to these standard and existing layers, we suggest to adopt in a similar way as semantic web services, another layer, called *Semantic-Layer*, which includes two sub-layers that are *Domain-Layer* and *Integration-Layer*. The *Domain-Layer* aims to describe and publish the three fundamental 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 specific ontologies and are the specialization of OWL-S (Web Ontology Language-Services). Concerning the *Integration-Layer,* it provides some technical services (mentioned above) in order to semantically discovery, mediate and execute fundamental services that are described and published by the layer above (*domain layer*). In the next section, some important technical services of the *Semantic-Layer* will be developed.

4.2 Semantic Layer Services

Semantic-Layer services are the main services that address the semantic problem.

They are divided into *Domain-Layer-Services* and *Integration-Layer-Services*. The most important technical service of each layer (which are *Description-Services* and *Mediation-Services)* will be described below.

4.2.1 Description Services

The principle of ODSOA is based on the use of some knowledge registries that store some formal ontologies, which are exploited by *Description-Services* in order to define the semantic description of services. According to Gruber, an ontology is defined as an explicit and formal specification of a conceptualization (Gruber, 1993), and for our purpose, we have defined three major types of ontologies: information or data-based ontologies, behavior or functional–based ontologies and process or business-based ontologies.

Data-based ontologies are the most basic ones. They provide semantic description of the data. Theses ontologies are required in all cases, no matter if we leverage functional-based or business-based ontologies.

Functional-based ontologies define semantic description around functions that are provided by the multiple EASs (and then services) 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 both business processes and applications.

Furthermore, *Description-Services* are based on the context of a service (*Service-Context*), which is defined by a set of ontologies, related to the concerned service and used for the annotation process. This *Service-Context* is also called local ontology, which means that there are several ontology levels. For our purpose, three ontology levels have been identified: *local level*, *domain level* and *global level.*

In essence, local ontologies concern services, whereas domain ontologies concern the generalization of local ones that belong to the same domain (Production, Metrology, Packaging, etc.) and they can serve in aligning the involved local ontologies. At last, global ontology is considered as generalization of domain ontologies, it is the root of the ontology hierarchy, and they can serve both in aligning domain ontologies and also in B2B integration that constitutes a natural prospect of our present work.

Our ontology architecture is somewhat an extension of the hybrid ontology approach mentioned in the case of information integration in (Wache, 2001). This extension is motivated by the fact that none of the approaches proposed by (Wache, 2001) (single ontology, multi-independent-ontologies and hybrid-ontology approach) are appropriate to fully capture and correctly structure semantics in our case.

This ontology clustering, which is firstly used in a general fashion in (Visser, 1999), is a very important concept in order to master the ontology evolution. We call this structuring *Ontology Urbanization* and it is related to the concepts of city-planning (zones, areas and islands) applied to EASs. It takes an important role in our integration approach and it will be more developed in future work.

4.2.2 Mediation Services

Mediation-Services are generally invoked by *Brokering-Services* (technical services that aim to provide global mechanism to integration process) in order to perform matching or resolution of semantic heterogeneity between *Fundamental-Services*. They exploit the description provided by the *Description-Services* described above.

Since we use an hybrid ontology approach, this requires the integration (mediation) of ontologies which are performed by *Ontology-Mediation-Services* (OMS) and that are based on ontology mapping (Kalfoglou, 2003). 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 (e.g. equivalence, subsumption), 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 mainly distinguish three other types of mediation services: Data Mediation Service (DMS), Functional Mediation Service (FMS), Business Mediation Service (BMS). These mediation services aim to mediate respectively between DSs, FSs, BSs and they are based on OMS that match and mediate between different ontologies. To be performed, *Mediation-Services* 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 (Racer, 2004) and OLA (OWL Lite Alignment) matcher (OLA, 2004) that seems be appropriate to our approach.

4.3 The initial Prototype

The initial prototype (also called ODSODI - Ontology-Driven Service-Oriented Data Integration), which is ongoing, aims to provide a first implementation of some functionalities of our architecture. We have restricted this first prototype to data integration. Further versions of the prototype will address application and process integration.

The underlying architecture of this first prototype is based around a fusion of WS concepts with the concepts of data mediation, especially the mediators concepts like those defined by (Wiederhold, 1992).

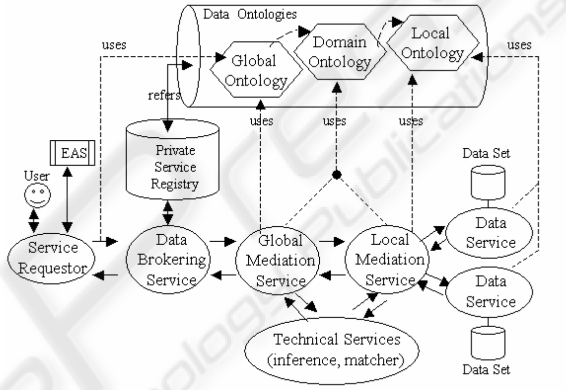

Figure 3: Principles of the initial prototype (ODSODI).

Figure 3 illustrates the principle of the ODSODI prototype. As shown, this prototype implements a local-centric approach (aka local-as-view approach) (Calvanese, 2001). In this approach, the query is done over the global ontology and the *Mediation-Services* access the data sets by a series of mappings: from global to domain (which are done by global mediation service), and then from domain to local (which are done by local mediation services). This choice is appropriate in the context of EAI in general and in the context of our microelectronics society in particular. It is motivated by the fact that users and EASs are autonomous and have a limited knowledge about DSs.

5 CONCLUSION

The semantic integration of enterprise application systems is a hard problem that can concern data, applications and processes. This problem needs ontology-based semantic mediation and, in our opinion, is best resolved in the context of service-oriented architectures.

This paper has proposed a unified approach for Enterprise Application Integration that exploits 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 some specific services that can mainly define the service semantics and also perform semantic mediation in the context of EAI. Typologies of services and also of ontologies have been suggested, and the initial prototype is described. This latter is of course limited, and its extensions, which may increase the field of use and the usefulness of our approach, will no doubt constitute important prospects in the future.

REFERENCES

Abiteboul S., Benjelloum O., Milo T., 2002. Web Services and Data Integration. *INRIA report*.

Amjad U., 2004. The Emerging Role of the Web for Enterprise, *Proceedings of the IEEE, 92(9), pp. 1420-1438*.

BIJOnline, 2004. www.bijonline.com.

Bussler C., 2003. The Role of Semantic Web Technology in EAI. *In the Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*.

Bussler C., 2004. Semantic Web Services. *ICWE'04*. Munich.

Calvanese D., De Giacomo G., Lenzerini M., 2001. Ontology of integration and integration of ontologies. *In Proceedings of the 2001 International Description Logics Workshop (DL2001), pp. 10-19*.

Cordoso J., Bussler C., Sheth A., Fensel D., 2002. Semantic Web Services and Processes. *Tutorial at Federated Conferences*.

Dogac A., 2004. Semantic Web Services, *ICWE'04*, Munich.

EBPML, 2004. www.ebpml.Org.

Erasala N., Yen D. C., Rajkumar T. M., 2003. Enterprise Application Integration in the electronic commerce world. *Computer Standards & Interfaces 25, pp. 69-82*. Elsevier.

Fensel D., Bussler C., 2002. The Web Service Modeling Framework WSMF. *In Electronic Commerce Research and Applications, 1(2)*. Elsevier.

Goh C. H., 1997. Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources. *Phd Thesis, MIT*. USA.

Gruber R. T., 1993. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *KSL, Standford University*.

Izza S. et al., 2004. Intégration d'applications: Etat de l'art et Perspectives. *JIE 2' 04 - Algérie*.

Kadima H., Monfort V., 2003. Les Web Services. *Dunod*.

Kalfoglou Y., Schorlemmer M., 2003. Ontology Mapping. *The Knowledge Engineering Review, Vol. 18(1), pp. 1-31*. Cambridge University Press.

Kontogiannis K., Smith D., O'Brien L., 2002. On the Role of Services in EAI. *In the Proceedings of the 10th International Workshop on Software Technology and Engineering*.

Linthicum D. S., 2004. Next Generation Application Integration. *Addison-Wesley*.

Lithicum D. S., 1999. Enterprise Application Integration, *Addison-Wesley*.

McIlraith S., Son T., Zeng H., 2001. Semantic Web Services, *IEEE Intelligent Systems, pp. 46-53*.

Mena E., Kashyap V., Sheth A., Illarramendi A., 1996. OBSERVER. *In Proceedings 1st IFCIS International Conference on Cooperative Information Systems, Brussels*.

METEOR-S, 2004. http://lsdis.cs.uga.edu/Projects/METEOR-S.

Noy N. F., 2004. Semantic Integration: A Survey Of Ontology- Based Approaches. *Standford Medical Informatics. Standford*.

OLA, 2004. www.iro.umontreal.ca/~owlola/alignment.html.

OWL-S, 2004. www.daml.org/services/owl-s/1.0/.

Racer, 2004. http://www.racer-systems.info.

Stuckenschmidt H., Wache H., Vogele T., Visser U, 2000. Enabling technologies for interoperability. *Workshop on the 14th ISCSEP, pp. 35-46. Germany*.

SWSI. 2004. Semantic Web Services Initiative, www.swsi.org.

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. *26th International Conference on Software Engineering (ICSE'04). IEEE Computer Society. pp 552-561*.

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*.

W3C, 2004. www.w3.org.

Wache H. et al.., 2001. Ontology-Based Integration of Information. *In Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing (2001) pp.108-117*.

Wiederhold G., 1992. Mediators in the Architecture of Future Information Systems, *IEEE Computer Society, 25(3), pp. 38-49*.

Woelk D., Tomlinson C., 1994. The InfoSleuth project. *In Second World Wide Web Conference '94: Mosaic and the Web*.