Workflow - Coordinated Integration of Enterprise / Industrial Systems based on a Semantic Service - Oriented Architecture

C. Alexakos\textsuperscript{1,2}, A.P. Kalogeras\textsuperscript{1}, S. Likothanassis\textsuperscript{2,4}, J. Gialelis\textsuperscript{3} and S. Koubias\textsuperscript{3}

\textsuperscript{1}Industrial Systems Institute  
Science Park of Patras,  
Stadiou Str, 26504,  
Platani, Greece  
kalogeras@isi.gr

\textsuperscript{2}Pattern Recognition Lab.  
Dept. of Computer Eng. and Informatics,  
University of Patras  
26500, Rio, Greece  
alexakos@ceid.upatras.gr

\textsuperscript{3}Applied Electronics Lab.  
Dept. of Electrical and Computer Eng.,  
Univ. of Patras, 26500 Rio, Greece  
\{gialelis,koubias\}@ee.upatras.gr

\textsuperscript{4}Research Academic Computer Technology Institute  
Kolokotroni Str, 26221 Patras, Greece  
lkothan@cti.gr

Abstract

Enterprise integration is significant for the enforcement of novel business models in an enterprise / industry. The great heterogeneity of systems / applications in the enterprise environment requires the introduction of interoperability aspects in order to resolve integration problems in a flexible and dynamic way. Our approach introduces an advanced enterprise semantic model representing both enterprise structure and available services, through the use of ontologies. The model is associated by a specific architecture that uses the above model in combination with state-of-the-art technologies such as Web Services and workflows.

1. Introduction

Modern enterprises need to follow an integrated approach regarding the opening up of their applications and systems and the exposition of their internal functionalities to their business partners. This approach is governed by three different aspects: the communication, the content and the required business logic. Communication related problems are dealt by via the wide acceptance of internet technologies and the spread of the internet infrastructure. Business logic regarding enterprise processes mandates the existence of a standardized framework providing integration capabilities. Finally, content is related to the introduction of semantics that may be associated with enterprise processes, products and resources and that represent in a uniform way the diverse enterprise functionalities.

Opening up enterprise processes may lead to vertical as well as horizontal integration. Vertical integration makes it possible for the different enterprise applications / systems to efficiently interoperate at whatever layer of the enterprise hierarchy they may reside in. Horizontal integration addresses the need for the interoperation of different plants of the same enterprise or different enterprises that may be viewed as business partners according to specific advanced business models, promoting collaborative B2B applications.

The current paper proposes a uniform integrated approach towards the semantic representation of an enterprise, introducing a semantic description of the enterprise structure as well as semantic annotations for the implemented enterprise Web Services. Dominant technologies are used such as Semantic Web Services, workflows and ontologies. An architecture is also proposed for the combination of the above technologies and our proposed approach implementation.

Section 2 presents the utilized state-of-the-art technologies. Section 3 is relevant to related work. Section 4 focuses on our approach and presents our proposed enterprise semantic model. Section 5 presents the associated architecture, while section 6 gives conclusions.

2. State of the Art Technologies

The proposed in this paper architecture is based on the utilization of three mainstream dominant technologies, more specifically semantic web services, ontologies and workflows. The combination of the aforementioned technologies provides a solution that may be characterized by a high degree of flexibility, interoperability and efficiency for the enterprise environment. The current section introduces these three technologies presenting the state-of-the-art.
2.1. Semantic Web Services

Web Services (WSs) [1] address a novel paradigm for the utilization of application inherent capabilities that are easily and dynamically exposed via open standard application interfaces and protocols. Such capabilities may be used by other applications through a simple WS invocation, resulting in extremely easy integration. In this context WSs also promote software reusability.

Although WSs add extreme efficiency regarding interoperability, it is still difficult for WS providers and requesters to have a straightforward and thorough understanding of nontrivial WS relevant statements such as inputs, outputs and constraints. [2] In order for this limitation to be alleviated semantic annotation is necessarily associated with WS description. Thus WSs may be transformed to Semantic WSs defined through their association with service ontologies.

The recognition of the advantages introduced by Semantic WSs has created a lot of activity in this area. A number of ontologies are proposed in the scientific community for providing a knowledge-based and conceptual description of the operations of web services. The starting point was made by D.Fensel et al [3] through the presentation of the Web Services Modeling Framework (WSMF), that provides the appropriate conceptual model for the development and description of web services and their composition using the DAML-S ontology [4]. In the recent years, such ontologies as OWL-S [5] and Web Service Modeling Ontology (WSMO) [6] were introduced for semantic markup of web services.

2.2. Ontologies

Knowledge representation is the driving force behind the first development of ontologies. A definition presented in [7] depicts an ontology as: “a hierarchically structured set of terms to describe a domain that can be used as a skeletal foundation for a knowledge base”. Ontology main feature is machine readability and understandability, resulting in the efficiency of different systems to easily utilize ontologies. This leads to increases platform independence as well.

The Semantic Web [8] introduction and unanimous approval has driven towards the representation of Ontologies in semantics. For this purpose a variety of semantic markup languages has been developed, based on the dominant XML standard. The most prominent ontology markup languages are DAML+OIL [9] and its successor OWL [10], which build on top of RDF Schema [11].

2.3. Workflow coordination

Enterprise processes may be identified through the coordinated invocation of different WSs implementing the various enterprise application functionalities. Their management is essential for the overall enterprise optimal operation which means that a strict control of their execution as well as their data flow is mandatory.

Workflows present a technology to achieve this supporting automation both inside an enterprise as well as among different enterprises. The above approach combining broadly accepted WSs and workflows, has mandated WS description in terms of workflow relevant standards. These include WSFL (Web Services Flow Languages) [12], XLANG [13] and Business Process Execution Language for Web Services (BPEL4WS) [14].

3. Related Work

Intra-enterprise integration and inter-enterprise integration is extremely significant for providing businesses a competitive advantage. Thus a lot of work has been done concerning these fields, mainly focusing on e-business applications and systems. Most of the relevant proposed implementations utilize the Service Oriented Architecture (SOA) according to which all involved applications and systems may be seen as components that may be instantly bound through the communication infrastructure, just at the time and for the time duration when they are needed. [15] Implementing SOA for WSs, Huang et al [16] proposed a service composition framework to support a Web services-based approach for developing e-business integration solutions enabling the composition of pre-defined canonical Web services.

Ontologies were introduced in the industrial domain by Schlenoff et al. [17] who proposed a two-tier ontology for manufacturing concepts and terms based on the Process Specification Language (PSL) utilizing a small set of primitive concepts like activity, object, time, point and relationship, and describing essential concepts with axioms. Furthermore, Hu et al. [18] propose an ontology based aspect integrator platform based on the IEC 61346 standard leading to a technical system model through objects, aspects and structures.

Some work focusing on enterprise integration solutions has been done regarding Web Service annotation through ontologies. More specifically, Patil et al. introduced the Meteor-S framework [19] for semi-automatically marking up Web service descriptions with ontologies. Furthermore, Tektonidis et al. [20] presented the ONAR framework, which provides the generation of semantic web services according to ontologies destined towards enterprise applications integration.

Enterprise system coordination using workflows that activate business processes through WS invocation has been used by Votis et al. [21] in the domain of enterprise grid networks. A similar work regarding the logistics domain has been done by Piccinelli et al. [22] introducing the DySCo framework. Work presented in this paper builds upon previous work presented in [23].
4. Proposed Semantic Enterprise Model

A widespread enterprise model, classifies its systems and their applications according to a three tier hierarchical model comprising the Enterprise, the Plant and the Shop-Floor Layer. Systems such as the Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) reside in the Enterprise Layer, the Plant layer is characterized by the Manufacturing Execution Systems (MES) while the lower Shop-Floor layer is associated with the manufacturing enterprise processes.

Communication technologies provide the backbone for the efficient unification of the above layers. More specifically internet ubiquity has given a boost to internet related standards and technologies, such as Web Services. In this context, enterprise systems at whatever layer they may reside in may expose their functionalities in a uniform way through the Web Service common interface. Furthermore, legacy systems is also possible to expose their functionalities in the same way through appropriate wrapping as for instance presented in [24].

In this context, enterprise activities of the different layers mentioned may be described using a uniform format according to the WSDL standard. The problem is nevertheless associated with the lack of knowledge about the role, exchanged data and tasks of enterprise activities, since their description following the WS model is restricted to binding and invocation information.

Thus, the utilization of WSs needs to be combined with semantic information, making it easier to alleviate the strict binding between an enterprise application and the actual enterprise infrastructure and systems it is implemented by or it integrates. This approach is expected to increase enterprise flexibility and component reusability.

Our work in this paper proposes the use of ontologies for the introduction of the above mentioned semantic information, related to industrial / enterprise operations and resources. The advantage of ontologies lies in the fact that they are machine-readable and machine-understandable, making it possible to conceptualize enterprise applications and systems.

In order to have an overall integrated view of the enterprise / industrial processes and resources through their representation using ontologies, it is essential to have a semantic description for both the enterprise structure and the web services provided by the enterprise / industrial systems and representing their functionalities. In this context, the roles, hierarchy and relationships of the different enterprise structural elements and systems need to be annotated through ontologies. Furthermore, the web services provided by the enterprise / industrial systems have to be semantically enriched with the intention of acquiring all the necessary information about their purpose, roles and parameter data of the operations they are associated with.

4.1. Enterprise Structure Semantics

Depending on the enterprise domain of activity, enterprise structure may differ substantially. These differences may be justified by the differences in the produced products or services, or the followed business model. This enterprise heterogeneity makes it mandatory to have more than one ontologies describing the different aspects of the entrepreneurial structure. This modular approach in enterprise semantic representation also contributes to ontology simplicity and ease of reuse in different enterprises. Our approach introduces two levels of semantic description, comprising a top-level ontology that describes the three aforementioned enterprise layers, and a set of domain ontologies that describe the structure, roles, enterprise / industrial systems and the data exchanged at each layer.

4.1.1. The Enterprise Ontology

Due to its generic nature, the top-level ontology between different enterprises is expected to present a large number of similarities. We call this ontology, shown in Figure 1, the “Enterprise Ontology” containing semantic information about the three layers in the enterprise. The Enterprise Ontology consists of three main classes the “Layer”, the “Department” and the “Job” which describe the main features that characterize an enterprise. Each of these classes contains properties and subclasses providing a supplementary description of the enterprise features.

The “Layer” class contains three subclasses the “EnterpriseL”, the “PlantL” and the “Shop_FloorL”. The Layer class is described by the properties: a) “Departments” denoting the departments that the layer consists of, which, in ontology terms, are instances of the class “Department” and b) “Details” which provides a text with the detailed description of the role of the layer in the enterprise.

The “Department” class consists of properties such as “Name”, “Role” and “Jobs” which are instances of the class “Job” and describes the role of the staff in each department.
The “Job” class describes the jobs that exist in an enterprise. The class illustrates properties like “Title”, “Description” and “Responsibility”. All three properties are text descriptions of the main characteristics of an employee in the enterprise.

4.1.2. The Domain Ontologies
The domain ontologies for each one of the three layers express its features. They refer to each layer of the specific enterprise elaborating the actual layer roles, jobs, enterprise / industrial systems and format of exchanged data.

The basic classes that are used in a domain ontology are the “LayerDetails”, the “DataExchanged”, and the “Systems”. There can be further classes, subclasses or properties in the main classes, in order to give more information about the enterprise structure and policy in the particular layer.

The “LayerDetails” consists of information about the layer that is described and the class purpose is mainly to provide the necessary mapping details to the top-level “Enterprise Ontology”. This class consists of subclasses such as “Layer”, “Departments” and “Jobs” which correspond to the classes of the Enterprise Ontology.

The “Data_Exchanged” and “Systems” contain subclasses and properties that are used to describe one or more types of data and enterprise / industrial systems correspondingly. An example of the Data_Exchanged subclass could be the “Product” class with the properties “Product_SerialNumber” and “Product_Name”. A Systems subclass could be the “PLC” class with such properties as “ModelName”, “DepartmentResiding”, “EmployeesUsers” and “ModelSerialNumber”.

The LayerDetails class is used for providing a definition of the layer that is described. The Data Exchanged class and the Systems class provide definitions of the exchanged data and the systems that reside in the industrial / enterprise environment.

4.2. Web Service Semantics
The Enterprise Ontology and the Domain Ontologies provide a concrete picture of the enterprise structure, enterprise / industrial systems as well as enterprise hierarchy. No reference is nevertheless made to the operations and services provided by the enterprise / industrial applications / systems.

According to our proposed approach, WSs provided by the enterprise / industrial systems are semantically described using the OWL-S ontology, in order to be annotated with additional information, such as description of their operation, mean of their argument data as well as their invocation details.

The OWL-S ontology consists of three types of semantic information, the Service Profile, the Service Grounding and the Service Model.

The Service Profile provides a machine understandable description of the operation that takes place upon a specific WS invocation. Thus, it is possible to convey such information as a description of the actual operation of the WS, limitations on service applicability and quality of service, as well as requirements that must be met by the service requester for the successful service invocation and use.

The Service Model provides the prospective service client a roadmap for using the WS, detailing the semantic content of requests, the conditions under which particular outcomes will occur, and, where necessary, the step by step processes leading to those outcomes.

The Service Grounding specifies details for the service client regarding the way to access a service through specific communication protocol, message formats, and other service-relevant details such as port numbers used in contacting the service.

Classic WSDL description of the web services conversion to OWL-S semantic description is possible through the utilization of such tools as ASSAM (Automated Semantic Service Annotation with Machine learning) [25].

Our approach makes it possible to use the semantic information of the OWL-S ontology for each different WS provided by the enterprise / industrial systems. This enterprise / industrial system functionalities may be utilized in a unified integrated way through their machine readable and machine understandable description.

4.3. Semantic Mapping
As presented above our approach introduces two different types of semantic descriptions: the former being associated with the enterprise structure and the latter with the WS characteristics. In this context, a complete description of the different enterprise aspects is possible. What is still missing is the association of the semantic information representing the structure of the enterprise with the semantics of the WSs that utilize or implement parts of this structure. This association promotes an integrated description of enterprise-relevant knowledge that can be machine-readable. A methodology is required to address the mapping needs between the different implemented ontologies providing the aforementioned semantic information.

The first step in this methodology concerns the mapping of the domain ontologies to the top-level Enterprise ontology. This mapping is possible by utilizing the instances of the LayerDetails class of each domain ontology and mapping them to the corresponding instances of the top-level ontology. LayerDetails mapping is related to such domain ontology details as departments and jobs, leading to an indirect association of the top-level ontology to such domain ontology characteristics as data exchange types and layer resident systems.

The second step in our methodology regards the needed mapping between different enterprise domain
ontologies. This mapping is necessary, since each domain ontology represents semantics of a different enterprise layer. Given that layer functionalities differ a lot, similar differences may be present to the corresponding ontologies. Since vertical integration needs require that information is seamlessly passed among the different layers, domain ontology mapping is absolutely necessary.

The final step in our methodology introduces the mapping of the domain ontologies to the semantic information related with the actual Web Services implemented at the associated layer. This mapping is related to information about the WS such as which system provides it, what are the request / response arguments as well as their type of data. This semantic information stems from the OWL-S ontology description of the web services and has to be mapped to the corresponding classes and properties of the relevant domain ontology.

This methodology is depicted in figure 2 providing an overview of the semantic information mapping, leading to the desirable unification of the top-level ontology with the domain ontologies and the semantically described web services, and thus contributing to the creation of a machine readable and machine understandable enterprise model.

5. Associated Architecture

The enforcement of the above presented enterprise model requires a specific architecture that makes it possible to integrate the different industrial / enterprise systems residing in different enterprise hierarchy layers in a flexible and interoperable way.

Our proposed architecture introduces a middleware named Coordination Middleware which role is to manage the aforementioned semantic enterprise model, as well as the needed enterprise process design, coordination / monitored execution through the application of workflows. Processes are thus broken down in workflow steps invoking Semantic Web Services, discovered through an OWL-S ontology empowered UDDI.

Coordination Middleware consists of four systems:
- The Ontology Data Base, which is a database management system where the top-level ontology as well as the domains ontologies are stored in OWL format.
- The Ontology Mapper presenting a tool that provides the user with the appropriate graphical interface for determining semantic mapping information.
- The Workflow Designer which is a tool with a simple and user-friendly interface for the composition of workflows using the semantic description of the enterprise.
- The Workflow Executor is a BPEL4WS execution engine which role is to initialize and execute an industrial process defined in terms of a workflow.

The integration consists of three discrete phases: a) the Mapping Phase, b) the Workflow Design Phase and finally c) the Workflow Execution Phase as shown in figure 3.

During the Mapping Phase, the Ontology Mapper is used in order to determine mapping information. The Ontology Mapper obtains the semantic descriptions from the Ontology Data Base as well as the UDDI server, and exports a XML-formatted file with all the defined associations. This XML file is stored in an internal repository in the Ontology Mapper.

The Workflow Design phase comprises the design of the enterprise process that has to be executed in the enterprise environment. The Workflow Designer gathers all semantic information from the Ontology Data Base
and the UDDI server as well as the mapping information from the Ontology Mapper and conveys it to the user in a human understandable way. User friendliness mandates this information to be shown in a seamless and transparent manner, meaning that no information regarding web service bindings or ontology terms is given. Instead, the user may design the needed workflow through the utilization of such enterprise aspects as roles, enterprise industrial systems, departments, types of data and activities. The designed workflow is also stored in an XML file in an internal repository of the Workflow Designer.

The Workflow Execution phase is activated when a demand for the execution of a industrial / enterprise process is placed. The Workflow Executor obtains the XML workflow file from the Workflow Designer and the required semantic information from the Ontology Mapper, the Ontology Data Base and the UDDI server and converts on-the-fly the workflow description in a BPEL4WS document. This document contains the necessary WSs that have to be invoked for the workflow execution, obtained according to the combination of the enterprise semantic model and the workflow. The BPEL4WS document is forwarded to an internal BPEL4WS execution engine where the execution of the process is initialized.

6. Conclusions

The proposed enterprise semantic model and the associated architecture provide an integrated approach towards achieving horizontal and vertical integration in the enterprise / industrial environment. Our proposal adds significantly to the overall system in terms of flexibility and efficiency while it introduces a high degree of system and application interoperability. The introduction of enterprise semantic information and the implementation of enterprise processes in terms of workflows invoking web services also increases the degree of enterprise component reusability.

The use of semantic information associated both with the enterprise structure and the available Web Services in an enterprise, is accomplished through the use of different types of ontologies leading to an enterprise semantic model. This model is the core for the overall proposed architecture, enabling the system engineer to design the required enterprise processes in a dynamic way taking into account enterprise semantic representation.

The usage of ontologies for describing the enterprise roles, systems, and processes provides a flexible way for adding and removing enterprise components, applications and systems, simply managing the corresponding semantic and mapping information.

References

[14] BEA, IBM and Microsoft. Business Process
Execution Language for Web Services (BPEL4WS) 1.1 (2003), http://xml.coverpages.org/bpel4ws.html


