Vertical Integration of Enterprise Industrial Systems Utilizing Web Services

A.P. Kalogeras¹, J. Gialelis², C. Alexakos¹, M. Georgoudakis², and S. Koubias²
¹ Industrial Systems Institute, Building A, University Campus, 26500 Rio Patras, Greece, e-mail: {kalogeras, alexakos}@isi.gr
² Applied Electronics Laboratory, Department of Electrical and Computer Engineering, University of Patras, 2650 Rio Patras, Greece, e-mail: {gialelis, georgoud, koubias}@ee.upatras.gr

Abstract

The need for interoperability is prominent in the industrial enterprise environment. Different applications and systems that cover the overall range of the industrial infrastructure from the field to the enterprise level need to interoperate. This quest is driven by the enterprise need for greater flexibility and for the wider possible integration of the enterprise systems. The current paper presents an architecture that utilizes three predominant state-of-the-art technologies, namely workflows, ontologies and web services in order to address the above quest in an efficient way.

1. Introduction

Modern enterprises are characterized by a need for greater flexibility and interoperability. A great number of systems and applications exist in the enterprise industrial environment covering the whole range of the enterprise system hierarchical model. This model comprises four different levers: the field lever, the shop floor lever, the plant level and the enterprise level.

The field and the shop floor levels comprise the applications that are associated with the industrial manufacturing processes. At these levels the industrial field level equipment, such as sensors and actuators, and the industrial controllers, such as programmable logic controllers and numeric controllers, are interconnected. Applications at these levels are relevant to the distributed control code that is needed for the industrial production processes.

The plant and enterprise levels cover applications that are traditionally characterized as upper layer applications. The information systems of the enterprise need to interconnect either at the level of a plant or in the broader sense of a distributed multi plant enterprise. Applications supported are for instance enterprise resource planning, order handling or warehouse management.

The ubiquity of the internet has led to the emergence of standards that allow the creation of a unifying infrastructure that makes possible the integration of the different systems at whatever layer they reside. There is nevertheless a dividing line between applications at the lower and the higher levels.

Our work presented in this paper is associated with a wider view that mandates the integration and interoperability of all the systems / applications that participate in the enterprise industrial environment independently of their level. An architecture is proposed for this purpose describing an enterprise process by means of a workflow, and utilizing the state-of-the-art technology of Web Services in order to open up industrial systems and make available their functionalities.

Our approach introduces a further degree of interoperability at the system engineering level by incorporating semantic information described in ontologies related with the enterprise processes.

The paper makes reference to the state-of-the-art technologies utilized by the proposed architecture, then it presents this architecture and finally it showcases an example process as well as its deployment according to our proposal.

2. State of the art

The architecture that is proposed in this paper is based on the utilization of three mainstream technologies, more specifically web services, workflows and ontologies. Their combination results in a scheme that promotes interoperability and flexibility at the system engineering level and addresses the needs of the industrial enterprise.

This chapter provides an initial introduction of the three technologies and presents the current state-of-the-art.

2.1. Web Services

Web Services (WSs) [1] represent a new paradigm for applications by utilizing capabilities that are inherent to and exposed by applications (or even other WS) via open standard application interfaces and protocols. An application can use the capabilities of a Web Service by simply invoking it across a network without having to integrate it. As such they represent reusable software building blocks that are URL addressable.

The capabilities provided by a WS can fall into a variety of categories, including mainly: function
activation, data transfer and business processes integration. Some of these capabilities are difficult or impractical to integrate within third-party applications. When these capabilities are exposed as WSs, they can be loosely coupled together, thereby achieving the benefits of integration without incurring the difficulties thereof.

Since, WSs expose to client applications their capabilities rather than their implementations, they can be implementation independent and still be compatible with all client applications. Their benefits include: reduction of application development costs, integration of both data and business processes, reduction of errors, simplification of application maintenance and significant reduction time-to-market. WSs are indeed a technology for distributed computing and there is one critical distinction between them and former distributed computing technologies since, a developer who implements a WS can be almost one hundred percent certain that everybody else can access and use this service.

The breakthrough of the WS technology is the anybody-to-anybody communication that it enables. WSs grew out of a request for a distributed computing application environment that would not be as difficult to deploy as CORBA or DCOM or JRMI and would also offer greater interoperability. All CORBA, JRMI and DCOM aimed to provide a distributed computing environment across heterogeneous environments. Unfortunately, neither of them supported communications at the anybody-to-anybody scale. The key enabler for WSs technology is XML and the initiatives that form its foundations are SOAP, WSDL, UDDI.

Simple Object Access Protocol (SOAP) [2] is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It is not tied to any particular transport protocol, although HTTP is the most popular. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics. This means that a client and a server can communicate as long as they can forward and understand SOAP messages. SOAP consists of the following parts: an envelope that describes the content of the message and how to process it, a set of encoding rules in order to express instances, application-defined data types and a convention for representing remote procedure calls and responses.

Web Services Description Language (WSDL) [3] provides a model and an XML format for describing Web services. WSDL enables the separation of the description of the abstract functionality offered by a service from concrete details of a service description such as "how" and "where" that functionality is offered. WSDL describes a Web service in two fundamental stages: an abstract one and a concrete one. According to the abstract description, a Web Service is given in terms of the messages it sends and receives. At the concrete level, a binding specifies transport and wire format details for one or more interfaces. An endpoint associates a network address with a binding. Finally, a service groups together endpoints that implement a common interface.

The last piece of the puzzle is UDDI (Universal Description Discovery & Integration) [4]. It supports both design-time and run-time discovery of Web Services.

2.2. Workflows

Process management enables full control of data during their routing through enterprise processes. Workflow management is the automated co-ordination and control of such processes. When a process is in a suitable form for being processed by a workflow management system (WfMS), it becomes known as a workflow. A workflow presents a model of an enterprise process that contains all the essential information needed to implement it. Such information may be: number of activities, activities order, data assignment, resources (material, people) designation, etc. In a more abstract sense, workflow systems tie together the three fundamental aspects of an enterprise: business processes, human resources and information technology (IT).

Given the broad acceptance of Web Services for enabling interoperability and their widespread support by vendors, they have become a natural choice for application integration within and across enterprises. Running workflows over such an interoperable system is attractive. Efforts are underway to define Web Services workflow standards. These include WSFL (Web Services Flow Languages) [5], XLANG [6] and BPEL4WS (Business Process Execution Language for Web Services) [7].

2.3. Ontologies

Ontologies came along in the area of Artificial Intelligence as a means of representing knowledge. Nowadays they present a topic of interest in diverse research areas and their usefulness can be perceived in a great number of applications. An exhibitory definition given in [8] is: “ontology is a hierarchically structured set of terms to describe a domain that can be used as a skeletal foundation for a knowledge base”. Ontologies give a specific description of a domain where its terms and their relationship is clearly defined. The terms are organized in a hierarchical structure and the relationships consist of HAS-A and IS-A associations. The main feature of ontologies is that the knowledge they describe can be noticeable from different users and be used for platform independent implementation.
By the evolution of the Semantic Web the need of the representation of the Ontologies in semantics, introduced a variety of semantic markup languages, based on the XML standard. The most prominent ontology markup languages are DAML+OIL and its successor OWL, which build on top of RDF Schema. It allows describing classes using description logics. Currently most ontologies use only a part of the feature set provided by the DAML language [9], thus many ontologies can be expressed in RDF Schema only [10], providing a basic ontological vocabulary to be used within a certain community.

3. The proposed Architecture

An enterprise process usually comprises a number of actions that are associated with the sequential use of different applications or systems ranging from the industrial field level up to the enterprise level. The need for interoperability of such systems / applications has led to a transition from closed legacy systems to more open ones, utilizing middleware technologies and the internet infrastructure for integration purposes. The architecture proposed in this paper builds upon this general trend, introducing a further step towards interoperability regarding the semantics of the involved applications / systems.

The very idea of the enterprise process as a sequence of different application / system service calls, directly introduces the idea of a workflow associated with it. A workflow is an abstraction of an enterprise process. It comprises a number of logic steps (known as tasks or activities), dependencies among tasks, routing rules, and participants. In a workflow, a task can represent a human activity or a software system. Most workflow standards support sub-processes, which allow tasks within a workflow to be implemented as another workflow. There are two complementary parts to a workflow: the control flow and the data flow. The control flow defines the sequencing of different activities in the process. The data flow defines how information flows between activities.

A workflow describing an enterprise process presents actually a means for expressing business logic in a standardized way. Given an enterprise process workflow, it is required that its tasks are associated with the diverse application / system services that must be called during the execution of the workflow, so that its incorporated business logic is successfully applied.

This paper emphasizes on the association of workflow tasks with web services that are available by the underlying applications / systems which compose the overall enterprise information infrastructure. As depicted in chapter 2, web services allow any piece of software to communicate with each other in a standardized XML messaging scheme. They eliminate many of the interoperability issues that the traditional integration solutions have difficulty in resolving. With minimal programming, the Web services technology enables the easy and rapid wrapping of legacy enterprise applications / systems and the exposure of their functionalities through a uniform and widely accessible interface over the enterprise communication infrastructure. In addition they can be mixed and matched to construct new and complete enterprise processes thanks to their standardized service interfaces and utilization of common communication protocols.

It would be sufficient in terms of making things function, to just provide an association of workflow activities with the available web services. This is nevertheless an approach that actually ties the process workflow to specific web services provided by specific systems / applications. Viewing it from the system engineering perspective, this approach lacks significantly in terms of flexibility and reusability, since the process workflow is very tightly connected to the actual enterprise infrastructure at a given instance of time. Thus, workflows would need significant alterations whenever changes in enterprise infrastructure take place, while on the other hand it is not easy to port existing workflows to different enterprises even in the case that processes involve the same or similar business logics.

In order to resolve the above issues and increase interoperability we propose the utilization of semantic information relevant to the different business processes. Business process semantic information may be introduced by means of defining ontologies. Ontologies emerged as a means to represent knowledge. Semantical relationships in ontologies are machine readable; thus they enable the creation of statements and the formulation of queries introducing a level of conceptualization, describing ontological entities and their relationships. In our present work, we propose the use of ontology semantics to represent roles, tasks and their relationships in an enterprise manufacturing environment, offering all the necessary information related to an enterprise process.

Our idea introduces an intermediate layer between workflows and web services. Enterprise process workflows are actually described in terms of the ontology associated with the processes. Thus an enterprise process is described by means of both its ontology and workflow.

The separation of the workflow and the services needed for the workflow execution through an ontological intermediate layer, increases the level of enterprise interoperability, since workflows need to be altered only whenever business logic is changed and are not influenced by enterprise information system alterations.

The actual web services that need to be called during the workflow individual step execution are hidden behind the ontology. It is nevertheless needed to
associate the ontological information with the available web services during the design phase.

The above architecture implementation involves the following functional elements:

The Workflow Design Tool which is the part of the architecture where the enterprise process business logic is expressed in terms of a workflow.

The Ontology – Web Service Association Tool which hides the actual web-services behind the enterprise process ontology.

The Workflow Execution Engine that is responsible for the implementation of the enterprise process business logic by executing its workflow.

The Enterprise Information / Industrial Systems that provide the web services that will be called at execution time.

The Ontology – Web Service Association Tool will have to incorporate the descriptions of the available web services. The tool consists of a) a UDDI registry which contains the description of the web services provided by the Enterprise Information / Industrial Systems and b) the Ontology which describes, in terms of classes and instances, the roles, tasks and the exchange of parameter data between business processes. The Web Services are described with WSDL documents in the UDDI registry while the Ontology is represented according to the RDFS specification. In this context, the Web Services can be understood as methods having input and output parameters. The Ontology – Web Service Association Tool outputs in an XML formatted document the association of the classes and instances of the ontology to the methods and parameters of the web services provided over the enterprise communication infrastructure. Thus an efficient mechanism is provided making the web services of an application / system easy to discover and associate with the business process semantic information contained in the ontology.

The Workflow Design Tool needs to communicate with the Ontology – Web Service Association Tool in order to get the process ontology and its associated web services. Its outcome workflow will be represented in BPEL4WS format in order to be provided as an input to the Workflow Execution Engine.

The Workflow Execution Engine is responsible for two main tasks, the execution of the web services described in each step of the workflow, and the monitoring of the progress of a workflow execution.

The proposed architecture consists of two parts, the Workflow Configuration and Design and the Business Process Execution, as depicted in figure 1.

In the Workflow Configuration and Design the first step is to associate the ontological information with the existing web services of the enterprise or industrial field level applications / systems. Using the Ontology – Web Service Association Tool the configuration XML documents are created which incorporate the web services description according to the semantic representation of the business logic elements. The
second step is to design the business process workflow. The Workflow Design Tool retrieves information from the XML configuration document, which is related to the implementation of the workflow tasks. Finally, a BPEL4WS document is stored in a workflow repository.

For the purposes of Business Process Execution, the Workflow Execution Engine accesses the workflow repository and retrieves the workflow corresponding to the business process that has to be executed. For all workflow tasks the engine calls in sequence the appropriate web services using the SOAP protocol for the interconnection of the enterprise applications / systems. When a web service is called it proceeds to the next task until the business process is completed.

4. Example Use Cases

The present chapter of this paper depicts an example case study in order to further elaborate the proposed architecture. The main goal of this architecture is the integration of the different systems / applications ranging from the enterprise to the industrial field level. A suitable scenario should involve a large number of enterprise systems and applications. Such a scenario could be associated with order placement and handling in a manufacturing enterprise.

4.1. Scenario Description

This scenario involves most of the departments of an enterprise, ranging from the administration to the manufacturing chain. In order to keep the description compact we provide just an overview of a simplified and reduced version of the business process that needs to be executed, when a new order is applied.

Upon the reception of an order by the sales department, an enterprise internal request is forwarded to the warehouse department. There, the availability of the ordered items is checked. In the case that the stock in the warehouse is not enough to satisfy the order, the enterprise manufacturing chain needs to be activated.

It is nevertheless mandatory to have an estimation of the availability of the resources involved in the manufacturing process before the manufacturing chain is activated. The term “resources” covers all three: raw materials required, production equipment that needs to be allocated to the specific production process for a specific period of time and the availability of human resources for the same period.

This estimation may alter the production plan of the enterprise using as a possible criterion the cost for / benefit to the enterprise. On the other hand, it may activate processes for the placements of orders for required raw materials.

Upon the certification of the availability of all resources, the actual manufacturing equipment needs to be appropriately configured / initialized in order to start a manufacturing process for the satisfaction of the placed order. This configuration step would mean the downloading of the necessary files to the different controlling devices, e.g. PLCs, NCs, the required initialization and the necessary programme invocations.

This scenario clearly requires the involvement of different systems being employed on different levels of the enterprise hierarchical infrastructure, and their integration so that interoperability and flexibility of the overall process is achieved.

4.2. Process Design Phase

The fulfillment of the above scenario utilizing the proposed architecture sets some prerequisites with reference to the needed infrastructure. First of all an enterprise communication infrastructure is required so that the different applications / systems can interoperate above it. This infrastructure will be based on standard protocols and the opening up of existing applications / systems via the Web Services technology.

During the design phase of the scenario, its business logic that is described in 4.1. has to be analysed into an ontology and a workflow. The ontology will provide all the semantic information concerning roles, tasks and their relationships of both the applications / systems and the humans involved in a process of ordering.

The workflow will have to define the sequence of actions needed in order for the business logic to be followed. This sequence of actions will be determined in terms of ontology tasks. The design of the workflow in terms of the scenario ontology is a function of the Workflow Design Tool.

Then, it is the role of the Ontology – Web Service Association Tool to associate the workflow with specific web services, found in the UDDI registry of the enterprise. The UDDI registry contains the WSDL descriptions of the web services of all the application / systems of the company.

The outcome BPEL4WS workflow description file contains all the necessary information that is relevant to the workflow as well as the real actions required by the participating systems / applications during the process execution phase.

4.3. Process Execution Phase

The base of the process execution is the Workflow Execution Engine, which uses the workflow BPEL4WS description in order to call the adequate web services from the different information systems existing in the enterprise.
A distinction can be made to the web services according to the level of the enterprise infrastructure that their application / system resides in. For the purposes of this paper, a classification into two categories could be made comprising the enterprise and the industrial level.

The enterprise level contains such applications as for instance the management of order / customers, the warehouse, the resource planning, and the handling of raw material orders.

The industrial level is associated with the applications that are needed for the actual production chain. These applications are usually not executed on PCs but are rather widely distributed among controllers and smart industrial field level equipment. Yet, their appropriate configuration has to take place beforehand.

The overall configuration of the manufacturing equipment could be exposed as a web service, responsible for the activation of the process of the appropriate control code deployment as well as its invocation.

In the above context, the proposed architecture provides an integrated approach for the handling of any kind of process in an enterprise, whether it is more bound to the upper or to the lower layers of the enterprise infrastructure. Thus it increases the enterprise interoperability and flexibility as understood from the perspective of system engineering and design.

5. Conclusions

This paper presented a proposed architecture for enabling interoperability of the applications / systems residing at all different levels of the enterprise industrial infrastructure. This architecture combines three state-of-the-art technologies: workflows, ontologies and web services. It provides a standardized way of describing business logic associated with an enterprise process. In this context, a process definition covers the definition of both an ontology and a workflow. The ontology introduces semantics related to the process while the workflow defines its execution steps. The association of the workflow with ontological information hides the actual applications / systems that need to be called, increasing thus interoperability at a system engineering level. A proposed tool takes care of the association of the ontology and the web services exposed by the different enterprise systems. The combination of the three above mainstream technologies has been showcased in a process example.

The work presented in this paper has been partially supported by the project CLIO – EB53 [11].

References

[1] W3C - Web Services Activity
http://www.w3.org/2002/ws/


