BPIMS-WS: A service-oriented Architecture for Trading Partners Integration

Giner Alor Hemández, José Oscar Olmedo Aguirre
Research and Advanced Studies Center of IPN (CINVESTAV)

Abstract

The benefits of connecting businesses through Web-services-based trading exchanges are huge. These changes allow various companies to connect supply chains in various industries across industries, introducing new efficiencies and ways of buying, selling, and brokering products and services. Connecting businesses through Web-services-based trading exchanges give companies the ability to transact in ways that were never before possible. These changes help connect buyers to sellers and provide a shorter procurement trading lifecycle. Having this account, in this work a business processes integration and monitoring system has been developed to automate, integrate and monitor many of the enterprise business processes described as Web services without recurring to large investments in software development and deployment. The contribution of this work consists in a service-oriented architecture that follows the SOA’s principles of improving economical benefits of business collaborations.

1. Introduction

Recently, the Service-Oriented Architecture (SOA) development paradigm has emerged to focus on radically improving the efficiencies of creating, modifying and extending solutions for enterprise application integration and process automation between organizations. SOA redefines the concept of an application from being an opaque procedural implementation mechanism to that of an orchestrated sequence of messaging, routing and processing of events. Web services are becoming the dominant technology for developing and deploying applications following the SOA’s principles so that the platform and language independent interfaces of web services allow the integration of heterogenous systems. Also, a growing number of commercial enterprises are redefining their business processes under this technology.

Therefore, business process management based on the SOA paradigm facilitates the design, analysis, optimization of business processes. It achieves this by separating process logic from the applications that run them, managing the relationships among process participants, integrating internal and external process resources, and monitoring process performance. Having this into account, we have developed a system named BPIMS-WS (BPIMS-WS stands for Business Processes Integration and Monitoring System based on Web Services) to enable the bading partners integration process. Furthermore, BPIMS-WS offers additional functionality for the dynamic integration of enterprises, discovery and invocation of business processes accessible as Web services.

BPIMS-WS provides a virtual marketplace where people, agents and trading partners can collaborate by using current Web services technology in a flexible and automated manner. To achieve this, BPIMS-WS offers mechanisms to publish and discover the business process provided by enterprises. Enterprises must register their business processes described as Web services to enable to potential clients to integrate their processes with them through BPIMS-WS. Furthermore, BPIMS-WS provide a set of graphic interfaces for publishing and discovering services and for monitoring the business processes involved in B2C/B2B collaboration. Finally, by using UML sequence diagrams, BPIMS-WS allows to visualize the SOAP based communication behavior of a business collaboration. All these features are described in the following sections.

The rest of this paper is structured as follows. In the next section we provide the main characteristics of BPIMS-WS. In the following sections we present the architecture proposed and discuss its design principles. Next, we describe a case of study for describing the functionality of BPIMS-WS. Then we review the related work in this area and emphasize the contributions of our work.
2. BPIMS-WS

BPIMS-WS contains an extended UDDI [1] node where commercial enterprises, services and products are registered. For the classification of business processes, products and services in the repository, BPIMS-WS uses broadly accepted ontologies such as NAICS [2], UNSPSC [3] and RosettaNet [4] respectively. In a similar way the functionality provided by a normal UDDI [1] node, Web services can be registered, published and discovered in BPIMS-WS.

Additionally, BPIMS-WS enable the composition of Web services. These types of Web services are created, instantiated and executed dynamically in a BPEL4WS [5] engine. Inside the composite Web services are the search of products based on their technical characteristics such as price, delivery time, stock quote, among others. Along with this market constraints, BPIMS-WS is able to perform complex activities such as buying for lowest price, better delivery time or distributed purchase, to mention a few.

Furthermore, BPIMS-WS comprises enterprise intra-workflow and inter-workflow Web services. Enterprise intra-workflow Web services are structured orchestrations of composite Web services that describe the internal activities developed as the intended behavior of an enterprise. Enterprise intra-workflow Web services are created and instantiated in the brokering system and executed by a WS-CDL [6] engine. Enterprise inter-workflow Web services describe the orchestration of the long-running conversation behavior of enterprise intra-workflows. Enterprise inter-workflow Web services are created and instantiated in the brokering system as explained in the next section and executed by a Federation of WS-CDL engines. A detailed explanation of these types of Web services is described in the following sections.

According to the emphasis on automation, BPIMS-WS can be accessed in two modes of interaction, either as a proxy server or as an Internet portal. In the first mode, BPIMS-WS can interoperate with other systems or software agents. Like a proxy, BPIMS-WS receives XML-encoded requests that are completed with business partner binding information, and then forwarded to the corresponding enterprise workflows. Eventually the responses to the requests are received back from the business partners’ workflows and then replied to the requesters. In the second mode, BPIMS-WS acts as an Internet portal that provides to the users a range of options among the Web services available through the brokering system. In this mode, BPIMS-WS presents to the users diverse GUIs to get access to the Web services provided. The architecture of BPIMS-WS is described in detail in the next section.

3. BPIMS-WS Architecture

The BPIMS-WS service-oriented architecture has a layered design following four SOA’s principles: (1) Integration, (2) Composition, (3) Monitoring and (4) Management. The basic functionality of BPIMS-WS is situated in the bottom layer, while the more complex functionality is situated in the upper layer. Like in other layered architectures, the purpose of each layer is to provide the access to the Web services required by the upper layers, hiding the details of how the Web services are implemented. The layers are abstracted in such a way that each enterprise workflow or software agent communicates with any counterpart of the upper layers. In this context, each layer has a defined function as explained in that follows. Fig. 1 shows the general architecture of BPIMS-WS. In Fig. 1 each layer has a defined function, which are briefly described next.

3.1. Web Service Layer

The business processes integration is provided in this layer. For doing this, we have developed an integration brokering service that allows to publish and discover Web services to make business processes integration. For the discovery of Web services, the integrating brokering service uses the extended UDDI node. A set of simple Web services are contained in the integration brokering service. The simple Web services consist of the following basic operations:

1) Web Services Registry are operations intended to store information in the extended UDDI node about: (i) potential businesses partners, (ii) products, and (iii) services.
2) Web Services Cancellation are operations deemed to remove from the extended UDDI node registered information concerning businesses, products and services.
3) Web Services Search are operations deemed to search for product technical information.
4) Web Services Meta-Information are operations intended to retrieve both services information and BPIMS-WS meta-information.

The structure and behavior of this layer can be understood with the following example.
Assume that a client is willing to find the technical information available about her preferred product. First, the client must select the type of the product she wants from a range of options offered through the internet portal (Step 1 in Fig 2). Then, BPIMS-WS obtains the request and formulates a query to the extended UDDI node (Step 2 and 3 in Fig. 2). The result to the query is a list of all the suppliers that have the requested product in their stocks (Step 4 in Fig 2). Then, for each one of these suppliers, BPIMS-WS formulates another query to the extended UDDI node to retrieve the corresponding URL that contains the Web service specification corresponding to RosettaNet’s PIP 2A5 (Query Technical Information). Once located the URL, BPIMS-WS builds requests for the invocation of the associated Web services to the enterprises found. Then, BPIMS-WS sends those requests to the enterprises and to obtain later on the responses (Step 5 and 6 in Fig 2). Next BPIMS-WS extracts the required information and builds a XML document. This XML document is presented in HTML using the Extensible Stylesheet Language (XSL) (Step 7 in Fig 2). The answer contains information concerning to the product (according to the invoked Web service) and the electronic address of the enterprise that offers that product. By means of using simple Web services, a client can get the price, the delivery time or the quantity available in stock of any registered enterprise in the extended UDDI node.

In summary, software agents and applications can get access to all the Web services that have been registered. The Web service layer of the architecture of BPIMS-WS is shown in Fig. 2.

32. Composite Web Services Layer

The support for composite Web services is provided in this layer. A composite Web service is the orchestration of several simple Web services. Composite Web services can be created at both design and execution time. The architecture of BPIMS-WS in the composite web services layer is shown in Fig 3.

For the execution of a composite Web service is firstly necessary to locate a suitable template from the BPEUWS repository that describes the intended commercial activities. The design of BPEUWS repository is presented in [7]. Under this schema, the templates are completely determined since commercial partners are known before hand.

In case of a purchase order scenario of books, the client might be interested in buying a book in the store that offers either the lowest price or the minimum delivery time.
If a client wants to buy several books at the lowest price, BPIMS-WS will retrieve from a database the location of the BPEUWS workflow template that uses the purchase-criteria selected (Step 2 in Fig 3). Once the template is located, BPIMS-WS uses the WSDL document and all related configuration files in order to instantiate them. BPIMS-WS obtains the templates that can be used to find the suppliers that offer the product required by the client. A query to a database containing the WSDL documents provided by BPIMS-WS can retrieve the appropriate Web services to obtain a number of pieces of commercial information like price, delivery time, quantity, and purchase access point of the product (Step 3 in Fig 3). The related WSDL documents are then analyzed, and all the relevant information is retrieved and used to complete the templates.

The first step to complete the templates is to include the namespaces of the services on top of the BPEUWS document identifying them with a number. A list of partnerLinkTypes is also included and a portType is placed within each partnerLinkType element. The portType name is divided into two parts: the namespace and the portType name of the remote service. The second step is to complete the executable BPEUWS document by including or modifying some elements.

After Web services namespaces, partnerLinks, input and output variables, and invocations to external services have been included, a namespace and a partner Link must be added for each Web service along with input and output variables. The instantiated templates are allocated in a BPEUWS engine for execution (Step 4 in Fig 3).

To communicate with the running workflow, BPIMS-WS builds SOAP messages containing the information provided by the client. Following our example, the client sends to the running workflow, the book code and the required quantity in a SOAP message. The workflow verifies also that the sum of all the quantities is at least the quantity requested by the client. If it is not true, an empty list is sent back to the client as response, which means that client’s request could not be completely fulfilled by any of the registered stores (Step 5 in Fig 3). Whenever the workflow has been successfully terminated, it sends back to the client the list of suppliers satisfying her conditions (Step 6 in Fig 3). Then, the workflow is deallocated from the workflow engine. After the client selects the suppliers, a BPWLAWS template for placing a purchase order is now retrieved from the repository, completed and executed as described before. By enacting this workflow, the purchase orders are sent to the suppliers and the corresponding answers from each supplier are eventually received.

We have shown so far only one example that illustrates the use of a composite Web service. However, a wide variety of other composite Web services involving some optimization criteria have also been developed and tested like minimum delivery time, distributed purchase, etc.

In [8] a far different approach to the problem of Web services composition is proposed. Executable BPEUWS documents are not created from a
repository of templates. Instead, a dynamic binder and invoker module communicates with a generic Web service proxy to dynamically bind external Web services. Despite the elegance of this approach, the generic Web service proxy becomes a bottleneck because it coordinates all kinds of interactions occurring in the business process, from finding suitable candidate services to binding compatible services and invoking them. In comparison, our approach in this layer produces separate workflows that run independently and more efficiently from each other without recurring to a central invoker.

In [9] mechanisms for dynamically discovering Web services using semantic extensions to UDDI are presented. In [10] a different approach is presented. They semi automatically generate process composition by using semantic capabilities of Web services. Finally, in [11] another approach is presented combining Semantic Web technology and BPEL4WS to achieve dynamic binding. In comparison to all these approaches, we have not considered any form of semantic matching to automatically select the suitable service required. Due to the serious difficulties in solving the problem of automatically finding the appropriate service, we left this responsibility to the customers, as they need to have more control in critical issues like this. Another contribution of this work is the proposal of defining generic templates of common business practices that are completed dynamically to bind partner information. Furthermore, we show the use of optimization algorithms (lowest price, minimum delivery time, parallel purchase) to demonstrate the capabilities of our approach. Whereas other related works make extensions to UDDI or BPEL4WS to bind dynamically external services, we believe the proposed solution albeit simple can be rather effective.

3.3. Enterprise intra-workflow Web service layer

This layer comprises a repository of WS-CDL documents [6] containing a set of enterprise intra-workflow Web services. A WS-CDL document represents an enterprise intra-worldflow. An enterprise intra-workflow defines the behavioral aspects and the dependency relationships among the diverse entities that constitute the enterprise. An entity can be a billing department or a marketing department. The enterprise intra-workflow Web services are orchestrations of composite Web services. This orchestration is based on the WS-CDL model. In order to orchestrate the interaction of two or more enterprise workflows, a similar approach to the one described in the Composite Web service layer is followed in this layer that consists on the instantiation of generic process descriptions obtained from the WS-CDL repository and their further execution in the WS-CDL engine.

3.4 Enterprise inter-workflow Web service layer

This layer comprises a repository of inter-workflows descriptions and a set of enterprise inter-worldflow Web services. An inter-worldflow of an enterprise federation defines the policies and commercial ideologies along some governmental regulations dictated on the enterprises that constitute the federation. A federation can be a group of diverse enterprises that synergistically collaborate in pursuing common or complementary goals. The enterprise inter-workflow Web services integrates enterprise intra-workflow Web services oriented to the satisfaction of society needs such as to reduce the consumption of non-renewable natural resources or to increase food production given a limited budget.

Under our architecture the discovery, creation, composition, deploying, management and monitoring of Web services are supported. Though our approach uses a centralized model in which the coordination relies on BPEIAWS, WS-CDL and Federation engines, a distributed version of a networked array of the mentioned engines is being considered.

In the next section, we describe how business processes descriptions can be monitored at execution time. This is one of the more relevant aspects of BPIMS-WS in relation to the deployment of business process.

4. Process Activity Monitoring in BPIMS-WS

BPIMS-WS offers facilities for monitoring Web services. This facility is offered in all the layers of the architecture. For the monitoring process, it is necessary to listen to the request/response SOAP messaging of Web service-based business collaboration. SOAP messaging identifies the participants and their communications during the long-running interactions involved in the collaboration. For this end, BPIMS-WS intercepts all SOAP messages to generate a UML sequence diagram from the information about the participants and the order in which the messages are exchanged.

For the monitoring of activities, a set of Java classes has been developed to represent a UML diagram in a SVG (Scalable Vector Graphics) representation that can be visualized in an SVG enabled Internet browser.
The exchange of SOAP messages during some kinds of business collaboration may be developed very quickly. Therefore, to avoid reducing the performance of the Web services execution, the dynamic generation of UML diagrams uses a buffered mechanism to deal with the fast pacing production of SOAP messages.

In the next section, we describe the statistics tracked on deployed Web services in BPIMS-WS.

5. Web Services Management in BPIMS-WS

As Web services become pervasive and critical to business operations, the task of managing Web services and implementations of our Web services architecture is imperative to the success of business operations. In this sense, a basic web services manager was developed with capabilities for discovering the existence, availability, health, performance, and usage, as well as the control and configuration of Web services provided by BPIMS-WS. For each deployed web service by BPIMS-WS, the following statistics are tracked:

- Current state
- Total number of successful invocations
- Number of failed invocations (request received the server but resulted in an exception)
- Average response/transaction time for successful requests

To illustrate the functionality of BPIMS-WS, we describe a scenario that integrates several products and services suppliers that it has already been implemented in BPIMS-WS.

6. Case of Study: A Trading Partner Integration using RosettaNet PIPs

The case of study describes how BPIMS-WS facilitates the discovery, integration and composition of Web services that are offered by some electronic components suppliers whose business processes are conducted by RosettaNet PIPs. Suppose the following scenario:

1. The suppliers sell on-line electronic components products. They have registered their products and their business processes as Web services in BPIMS-WS.

2. A potential client (enterprise) has a supply chain and needs to find a products set to request a purchasing order by using Web services.

In this scenario, how can an enterprise integrate their business processes to locate and invoke the Web services available in BPIMS-WS to buy some products?

BPIMS-WS offers the modality of interaction as an Internet portal. In this mode, there is an option in the main menu called "Distributed Shopping". In this option, BPIMS-WS deploys a graphic interface where the clients can select some products registered and their respective quantities that want to find. The graphic interface of products selection is shown in Fig 4.

Once selected the products list, BPIMS-WS displays another graphic interface where the client must choose a sorting criteria. This sorting criteria indicates the form in BPIMS-WS will display the search result. Several criteria are lowest price, minimum delivery time, highest price, maximum delivery time, lowest price and minimum delivery time, and all their combinations.

Next, BPIMS-WS built a request to get_sortedProductsList composite Web service. This request returns a list of suppliers that have that product and satisfy the sorting criteria. The result is showed as a HTML document. At this point, a list of enterprises may appear as the product suppliers. Fig 5 shows the graphic interface with the result of the invocation to get_sortedProductsList composite web service. Next, the client selects an enterprise from this list where he wants to buy a product. Once did it, BPIMS-WS makes a query to the extended UDDI node to locate the URL where the PIP 3A4 (Request Purchase Order) is located and analyzes the Web service specification.
At this point, BPIMS-WS displays a graphic interface of the Web service specification, so that the client can visualize the activities involved in the purchasing order process. The client is asked to provide the information required to make the invocation of the Web service. In Fig. 6, the client can observe the SOAP messages involved in the business interaction.

Even for each generated UML sequence diagram that represents the business collaboration (in this case a RosettaNet PIP), BPIMS-WS can display its XMI representation to be portable in an UML editor. For this end, we developed a XMI viewer to transform UML sequence diagrams (described in SVG representation) to its XMI representation corresponding. Fig. 7 shows the XMI representation of a RosettaNet PIP described as UML sequence diagram in BPIMS-WS.

Upon completion, BPIMS-WS invokes the Web service. Finally, BPIMS-WS shows the HTML documents containing the answer.

We have shown in this paper only one example that illustrates the trading partners integration in BPIMS-WS. However, a wide variety of other cases of study involving shopping and using or not optimization criteria have been developed and tested such as shopping with the minimum delivery time, lowest price, specified quantity, without constraints, etc.

7. Related Works

A different approach to the one addressed in this paper is proposed in [12]. A P2P-based orchestration model to support the composition of multi-enterprise Web service is used. Instead in BPIM-WS, a layered architecture to provide support for discovering, creating, composing and deploying Web services was adopted in the P2P model. Our approach follows a client-server model, delegating the task of coordinating elementary activities to a BPEL4WS engine. Besides, monitoring of Web service is a unique feature of our approach.

A framework that builds on current standards to define extended service models and richer Web services abstractions is proposed in [13]. The main feature of the framework is a conversation meta-model derived from an analysis of e-commerce portal sites. Our approach has not considered any form of semantic description of Web services, but the interactions among business partners are determined on market constraints as price and delivery time.

A methodology for business process choreography is proposed in [14]. The methodology focuses on two
types of business processes (contract and executable) and provides an interface protocol to represent interoperability patterns between them. Procedures to incorporate existing workflows and generate collaborative processes are defined. BPML, BPEUWS and ebXML are used to define logical and internal processes. This work concerns to level three of our architecture. In comparison, our approach uses WS-CDL for describing any type of Web Service participant regardless of the supporting platform or programming model used by the implementation of the hosting environment.

8. Future Work and Conclusions

So far the design and implementation of the first two layers of BPIMS-WS is completed. As future work, we can mention the design and implementation of the WS-CDL engine. It is necessary for developing and deploying enterprise intra-workflow Web services. In enterprise inter-workflow layer, we need to define mechanisms to describe the set of internal behaviors of an enterprise federation. These mechanisms should be able to identify the relationships and interactions among the participants involved. It becomes necessary the design and development of an XML-based language and model that provides these features, along with its inference and execution engine. With this, we will design and develop the set of enterprise inter-workflow Web services corresponding to this level. Furthermore, we have considered a conceptual model to extend our architecture to effectively manage conversations between business processes and leverage web services from mere software components to actual entities of a business interaction [15].

In this work, we have proposed a service-oriented architecture for trading partners integration. The proposed architecture is based on four layers, where the basic functions are situated at the bottom layer where as the complex functions are situated at the upper layer. Furthermore, we define the functionality of each layer in the architecture. Finally, we developed a system named BPIMS-WS where the concepts and ideas of the architecture are described. BPIMS-WS provides access by means of an Internet portal where a user can appreciate the benefits of intermediation, integration and monitoring in trading partners integration.

9. References


[4] RosettaNet, RosettaNet Homepage


