BPIMS-WS: Service-Oriented Architecture for Business Processes Management

Giner Alor Hernández, José Oscar Olmedo Aguirre

Research and Advanced Studies Center of IPN (CINVESTAV).
Electrical Engineering Department. Computing Section.
Av. Instituto Politécnico Nacional 2508, Col San Pedro Zacatenco. 07360 México, D. F.
e-mail: gineralor@computacion.cs.cinvestav.mx, oolmedo@delta.cs.cinvestav.mx

Abstract. Service-Oriented Architecture (SOA) development paradigm has emerged to improve the critical issues of creating, modifying and extending, solutions for enterprise application integration, process automation and automated exchange of information between organizations. Web services technology follows the SOA's principles for developing and deploying applications. Besides, Web services are emerging as the platform for service-oriented architecture (SOA), for both intra- and inter-enterprise communication. 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. Our service-oriented architecture is based on a layered design that meet crucial design aspects like abstraction, scalability and interoperability.

Keywords: BPEL4WS, Business Process Integration, Service-Oriented Architecture, Web Services.

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 (Business Processes Integration and Monitoring System based on Web Services) to enable the integration, composition and monitoring of the processes developed by the business partners involved. Furthermore, BPIMS-WS offers additional functionality for the dynamic integration of enterprises, discovery and invocation of business processes accessible as Web services.
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 a 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 like NAICS [2], UNSPSC [3] and RosettaNet [4]. In a similar way to the functionality provided by a UDDI [1] node, Web services can be registered, published and discovered in BPIMS-WS. This UDDI node is the main component of our service-oriented architecture. As Web services proliferate inside the enterprise, there is a need to advertise, discover and reuse services, and UDDI is the standards-based mechanism for doing this. 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. Also, 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. We have considered the design of Enterprise intra-workflow Web services so that they can be created and instantiated in the third layer of our service-oriented architecture 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. In a similar way, Enterprise inter-workflow Web services have been considered so that they can be created, instantiated and executed in the fourth layer.
The BPIMS-WS service-oriented architecture has a layered design following three SOA's principles: (1) Integration, (2) Composition, and (3) Monitoring. The business processes integration is offered in the first layer of BPIMS-WS architecture. The creation of new Web services through the composition of existing business processes described as Web services is offered in the second layer of BPIMS-WS architecture. The monitoring business processes is provided starting from second layer. In the following sections, we describe with more detail the functionality of each layer of our service-oriented architecture proposed.

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 Internet portal is executed on a Tomcat application server.

3 BPIMS-WS Architecture

The BPIMS-WS service-oriented architecture has a layered design. 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 well-defined function, as it is 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 UDDI node. A set of simple Web services are contained in the integration brokering service. The set of simple Web services consults the information stored in the UDDI node using an ODBC/JDBC compliant database. The simple Web services consist of the following basic operations:
1) **Web Services Registry** comprises operations intended to store information in the UDDI node about: (i) potential businesses partners (businessName, description, discoveryURL, contactName, phone and e-mail), (ii) products (productCode), and (iii) services (serviceName, description, accessSOAP, accessWSDL and serviceCode). Examples of these Web services are save_Business, save_Services and save_Products.
2) **Web Services Search** consists of operations deemed to search for product technical information. Examples of these Web services are: get_Quantity, get_Price,

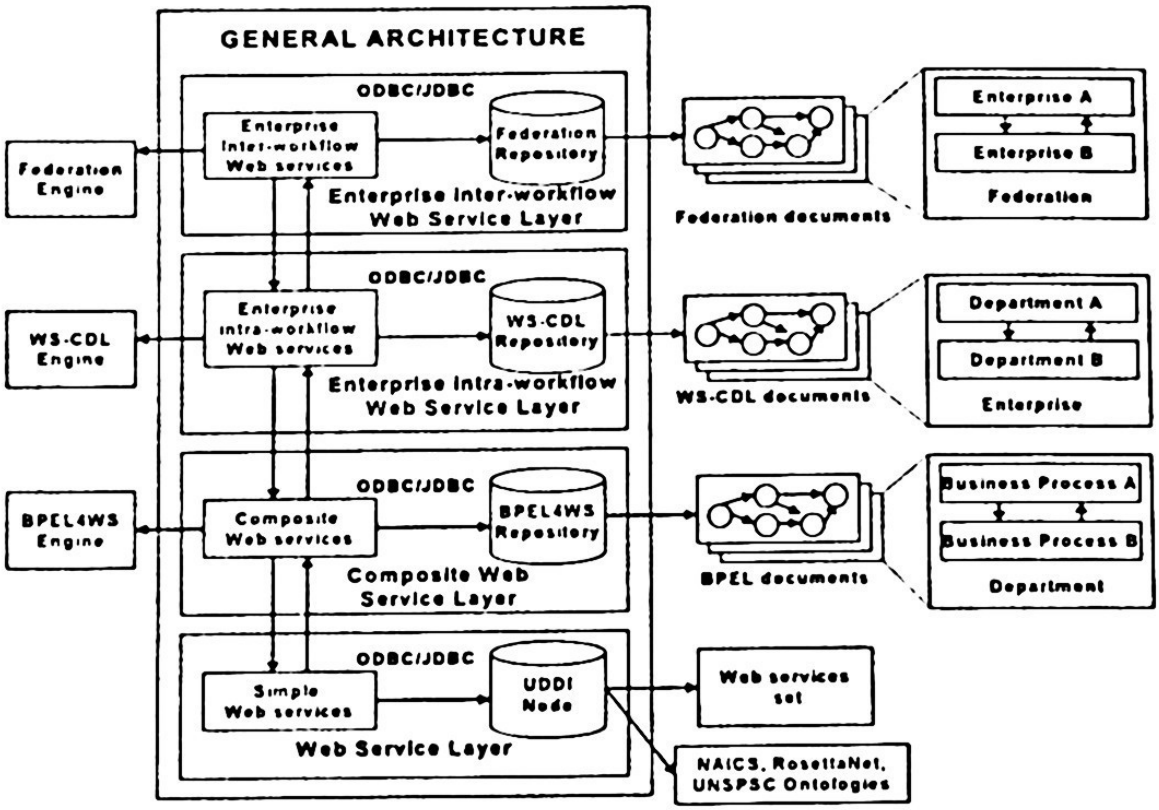

Fig. 1 General Architecture of BPIMS-WS

3) Web Services Meta-Information includes operations intended to retrieve both services information and BPIMS-WS meta-information. Examples of these Web services are get_BusinessOntology, get_ServicesOntology, get_ProductsOntology, get_ServicesList, get_RegisteredBusiness, find_Product, find_Service, find_Business and get_PasswordRecovery [7].

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 UDDI node. The result to the query is a list of all the suppliers that have the requested product in their stocks (Step 2 in Fig 2). Then, for each one of these suppliers, BPIMS-WS formulates another query to the 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 3 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 4 in Fig 2). The answer contains information concerning to the product (according to the invoked Web service) and the electronic address (discoveryURL) 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 UDDI node. Additionally, BPIMS-WS can also work based on more elaborated searching criteria [8]. 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.

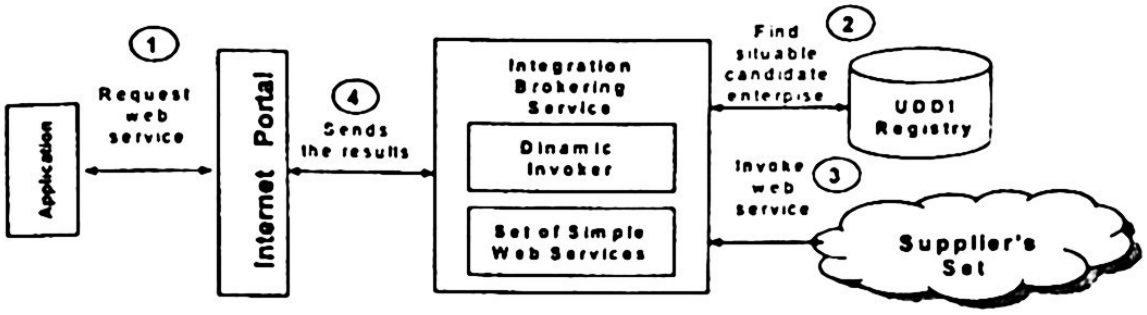

Fig. 2 Architecture of BPIMS-WS in the Web Services layer

3.2. 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.

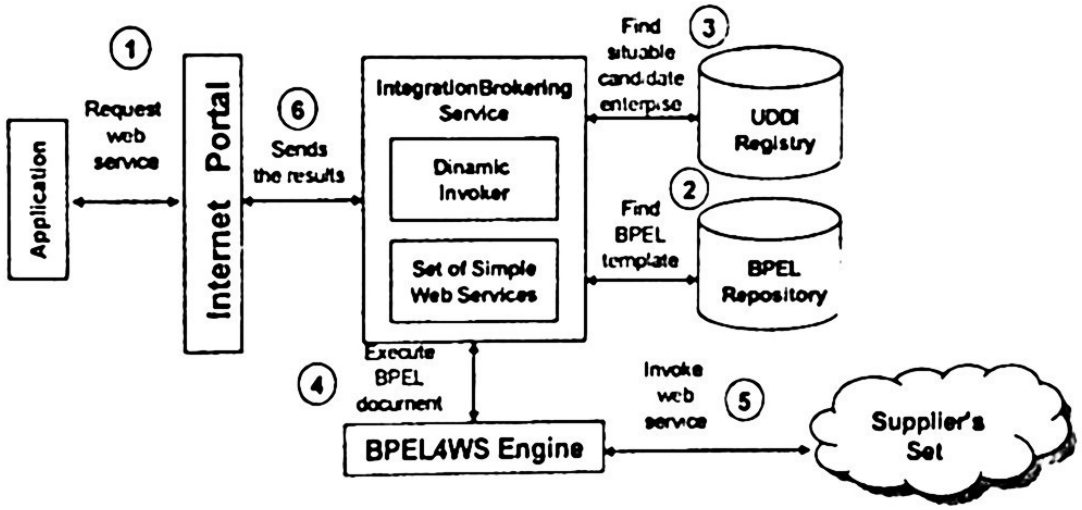

Fig. 3 Architecture of BPIMS-WS in the Composite Web services Layer

For the execution of a composite Web service is firstly necessary to locate a suitable template from the BPEL4WS repository that describes the intended commercial activities [9]. In this schema, the templates are completely determined since commer-

cial partners are known before hand. Next, BPIMS-WS retrieve from a database the location of the BPEL4WS workflow template that uses the commercial activity (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 products required by the commercial collaboration. 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 products (Step 3 in Fig 3). These services, based on UNSPSC and RosettaNet ontologies, are get_PriceandDeliveryTime, get_ProviderQuantity, and get_ProviderURLBuy, respectively. The related WSDL documents are then analyzed, and all the relevant information is retrieved and used to complete the templates.

The instantiated templates are allocated in a BPEL4WS 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. Here, the client sends to the running workflow, the information necessary to run the workflow such as product code and the required quantity in a SOAP message. The workflow verifies the constraints and pre-conditions and then it is executed. (Step 5 in Fig 3). Whenever the workflow has been successfully terminated, it sends back to the client the list of suppliers satisfying the conditions (Step 6 in Fig 3). Then, the workflow is de-allocated from the workflow engine. After the client selects the suppliers, a BPWL4WS 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.

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.

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-workflow. 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-workflow Web services. An inter-workflow 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.
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.

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. The right hand side of each one of the screenshots of Fig. 4 shows the UML sequence diagram of the business processes.

5 Case of study: A Supply Chain Management

The case of study describes how BPIMS-WS facilitates the discovery of Web services that are offered by some enterprises that sell electronic components. Suppose the following scenario:
1. The enterprises sell on-line electronic components products. The enterprise has registered their products and their business processes as Web services in BPIMS-WS.
2. A potential client (enterprise) needs to request a purchasing order by using Web services.

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

BPIMS-WS offers two modalities of interaction, either as a proxy server or as an Internet portal. We will show the integration process in the Internet portal mode. Some screenshots of the internet portal mode are shown in Fig 4.

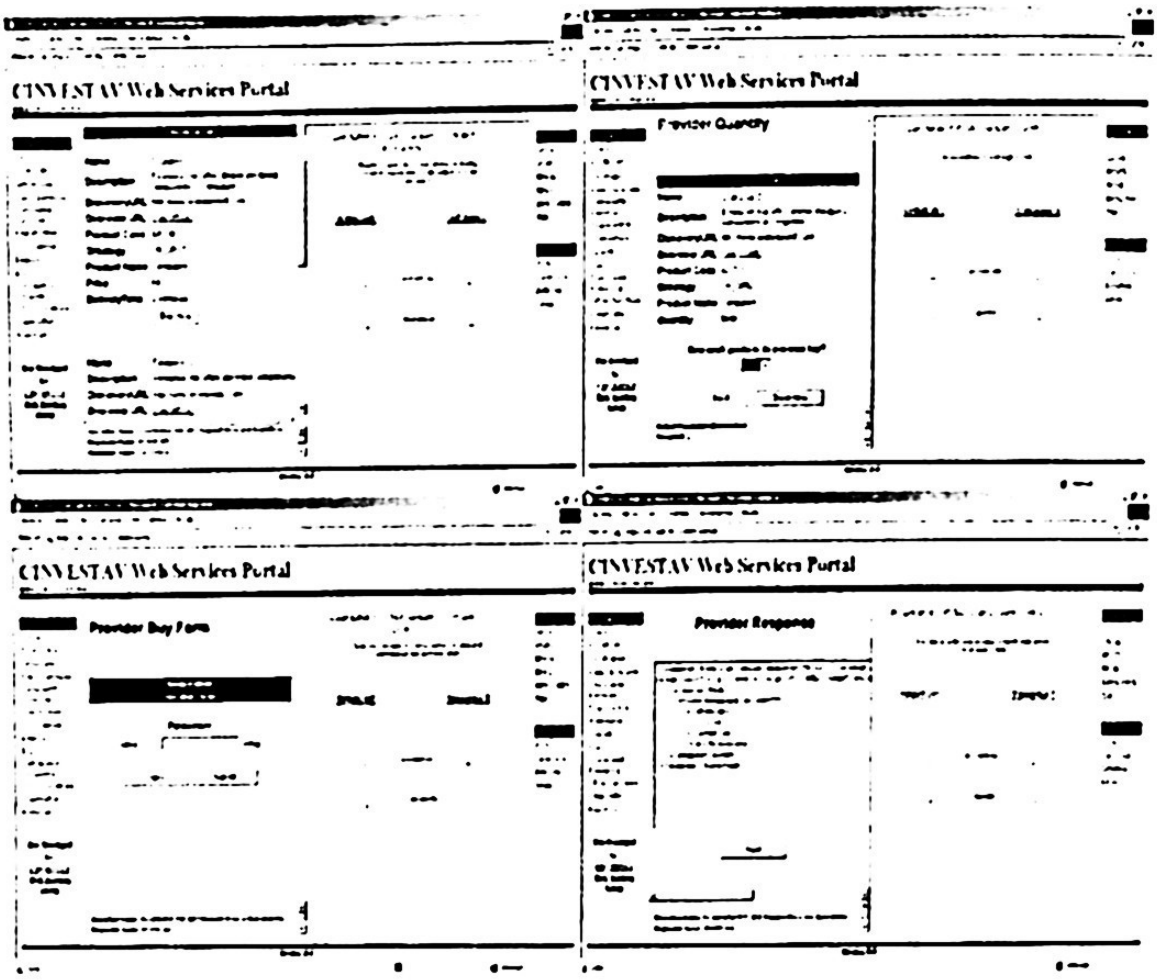

Fig. 4 Screenshots of BPIMS-WS in Internet portal mode

In the Internet portal mode, there is an option in the main menu called "Buy". In this option, BPIMS-WS deploys a graphic interface where the clients can make the search of some product registered. Then, the client can select the purchasing criteria. Suppose that the client has selected the "Show all providers" criteria. Once selected, BPIMS-WS consults the UDDI node to find all the product suppliers [10]. This query returns a list of all the suppliers that have that product. Next, BPIMS-WS retrieves from the UDDI node the URL where the RosettaNet's PIP 3A2 (Request Price and Availability) is described as a Web service for each one of these suppliers (upper-left picture in Fig. 4). Then, BPIMS-WS makes an invocation to each one of these Web services and obtains the corresponding answers. A list of results is obtained as an HTML document [11]. At this point, a list of enterprises may appear as the product suppliers. After the client selects an enterprise from this list, BPIMS-WS formulates a query to the UDDI node to find the URL where the PIP 3A1 (Request Quote) is de-

scribed [10] (upper-right picture in Fig. 4). Next, BPIMS-WS invokes the Web service and obtains the corresponding response. Similarly, the response is shown as an HTML document [11]. Here, the client should select the quantity of products that she wants to buy according to the product availability in stock. Once selected the number of products to buy, BPIMS-WS makes a query to the UDDI node to locate the URL where the PIP 3A4 (Request Purchase Order) is located and analyzes the Web service specification (lower-left picture in Fig. 4). At this point, BPIMS-WS deploys 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. Upon completion, BPIMS-WS invokes the Web service. Finally, BPIMS-WS shows the HTML documents containing the answer (lower-right picture in Fig. 4).

6 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 methodology for business process choreography is proposed in [13]. 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 [14], BPEL4WS and ebXML [15] 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.

7 Future Work and Conclusions

So far the design and implementation of the first two layers of BPIMS-WS is almost complete. 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.

In this work, we have proposed a layered architecture for the brokering of business processes for B2B e-commerce. The proposed architecture is based on four layers, where the basic functions are situated at the bottom layer where as the complex func-

tions are situated at the upper layer. Furthermore, we define the functionality of each layer in the architecture. Also, the architecture follows design principles as interoperability, integration, abstraction and scalability. 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 e-commerce B2B.

References

1. UDDI, UDDI Version 3.0, Published Specification, July 19, 2002.
2. North American Industry Classification System, NAICS Homepage, http://www.naics.com/.
3. United Nations Standard Products and Services Code, UNSPSC Homepage, http://www.unspsc.org/.
4. RosettaNet, RosettaNet Homepage, http://www.rosettanet.org/
5. Business Process Execution Language for Web Services. BPEL4WS 1.1 Specification. IBM May 5 2003.
6. Web Service Choreography Description Language. WS-CDL Specification 1.0. WS-CDL Home, http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/
7. Giner Alor Hdez, José O. Olmedo Aguirre. *Sistema de Intermediación para el Comercio Electrónico basado en servicios Web.* Proceedings CIC 2003.
8. Giner Alor Hdez, José O. Olmedo Aguirre. *Búsqueda, Localización e Invocación Dinámica de Servicios Web utilizando WSIL.* Submitted in CNCIIC-ANIEI 2004.
9. César Sandoval Hdez, Giner Alor Hdez, José O. Olmedo Aguirre. *Dynamic generation of organizational BPEL4WS workflows.* Proceedings ICEEE-CIE 2004.
10. Giner Alor Hdez, César Sandoval Hdez, José O. Olmedo Aguirre. *Descubrimiento Dinámico de Servicios Web en nodos UDDI mediante USML.* Proceedings CORE 2004.
11. César Sandoval Hdez, Giner Alor Hdez, José O. Olmedo Aguirre. *Generación Dinámica de GUI's para la invocación de servicios Web publicados en nodos UDDI.* Proceedings CORE 2004.
12. Boualem Benatallah, Quan Z. Sheng,, Marlon Dumas. "The Self-Serv Enviroment for Web Services Composition". IEEE Internet Computing January-February 2003 pages 40-48.
13. Jae-yoon Jung, Wonchang Hur, Suk-Ho Kang, et. al. "Business Process Choreography for B2B Collaboration". IEEE Internet Computing January-February 2004 pages 37-45.
14. Ismael Ghalimi & Jeanne Baker. BPML 101: Implementing the BPML Specification. BPMI.org Board Members, March 2002
15. Ebxml, Ebxml Business Process and Business Information Analysis Overview. Ebxml Business Process Team. May 2001.