

On-Demand Business Process Integration Based on Intelligent Web Services

Xiaohua Lu¹, Yinsheng Li¹, Ying Huang²

¹Software School, Fudan University, Shanghai, China

Phone: +86-21-55664096-808, {0014010, liys}@fudan.edu.cn

²IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

yxh@us.ibm.com

ABSTRACT

This paper proposes to make On-Demand business process integration based on intelligent Web Services. Business processes are modeled with BPEL4WS. They are incorporated into intelligent Web Services as their knowledge. Implementation framework and techniques are presented. A case of online auction is developed to demonstrate these concepts.

Keywords: on-demand business, process template, intelligent Web Service, BPEL4WS, online auction

1. INTRODUCTION

Businesses are becoming customer-centric from product-oriented. Enterprises have to respond with speed to customer demands, market opportunities or external threats. They therefore develop their e-Business systems to be more responsive, variable and resilient, i.e., on demand. To this end, their electronic business processes are required to integrate end-to-end across the company and with key partners, suppliers and customers. Service-oriented architecture (SOA) provides a competent computation environment for enterprises to integrate their applications and become on demand. Businesses can become flexible through integration of people, processes and information within and beyond enterprises. SOA and standardized Web services technologies make it possible to simplify information systems through automation and virtualization, enable access to and create a consolidated, logical view of resources across a network. Every application and resource is represented as a service with a standard interface which simplifies its activation and the exchange of information. Enterprises can treat business processes and the underlying infrastructure as defined components that can be mixed and matched at will. ^[4, 5]

To this end, numerous research efforts have been put on SOA based business process integration. This paper describes process template-based business integration, which is an ongoing research at IBM Watson Research Centre. We locate characteristics and advantages of template-based on-demand business, and try to improve it with intelligent Web services, which was proposed and studied in our previous publications. ^[1] The proposed intelligent Web services incorporate possible business processes, which are modeled with BPEL4WS, into its knowledge. They run as composition services and flexible business adapters in SOA-based application integration environments. The knowledge makes them rational to response to on-demand business or transaction requests with appropriate processes and call appropriate services. This will be demonstrated through a case study

of online auction for e-Marketplaces, in which an auction service is implemented with intelligence and can instantiate an appropriate process based on the requester's inputs, to accomplish an auction session.

2. TEMPLATE-BASED ON-DEMAND BUSINESS INTEGRATION

Today's e-business systems have been found with several drawbacks. An e-Business solution is usually created for a single enterprise, not flexible for an extended business. Systems of suppliers, manufacturers, sellers and customers are all developed with different technologies, running on different platforms. It is very difficult and costly when integrating systems of the whole supply chain. Also, it is not able to handle changes quickly and effectively when being required to meet requirements by another enterprise because business processes are buried in program codes. When processes changes, all these codes have to change. This is time-consuming and may lead to loss of potential business opportunities. Under on-demand business environment, all system components are services, and the whole system is constructed with service-oriented architecture. Some basic services are provided by platforms and service providers, while other business specific ones are created by enterprises own. That is, enterprises only need to focus on their business issues. We can create e-business solution by developing service components and defining templates to integrate these services. With this method, we can integrate e-business systems of different partners and create business solutions easily. When defining templates, business partners across supply chain are all incorporated. As long as they provide corresponding services, business process can be easily integrated with fewer costs than before. All these services and templates are reusable components. Moreover, business processes are no longer buried in codes. We rewrite those templates rather than the entire application to meet process changes.

In template-based on-demand business integration, there

are two concepts with particular importance, which are services and templates. Services comprise of business services and IT services. Business services can be further categorized into four types: demand services, such as sales; supply services, such as logistics; administrative services, such as human resource; control services, such as finance. IT services also have four categories: adapter services, such as data adapter; composition services, such as process flow; platform services, such as messaging; common services, such as authentication and authorization. Among these services, some can be found at service providers, while others need to be developed by enterprises. [2, 3]

When enterprises have all service components for e-business systems, they define business templates to compose those services into business processes and applications. Templates describe how services work with each other and how business processes execute. Composition services are used to integrate other services according to templates. Figure 1 depicts the framework of e-business solution using services and templates.

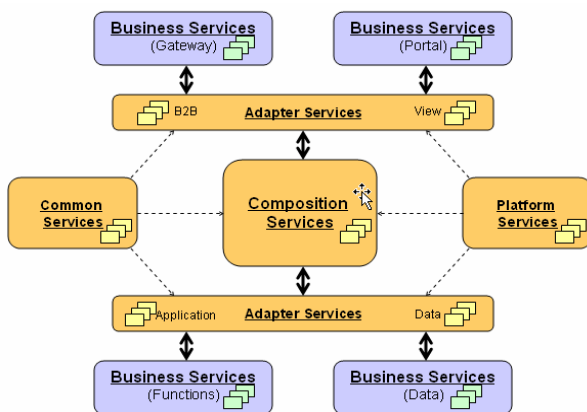


Figure 1. Framework of e-business solution using services and templates [2, 3]

In this framework, business services provide business functions. Common services and platform services provide facilities required by every application. Adapter services facilitate interaction between business services by providing agreed interfaces for them. And composition services integrate all the services above to form process templates and e-business system.

Template-based e-business solution has following advantages:

1. Process templates are independent of technologies and platforms. Changes in them will not impact existing system.
2. Templates and services are all individual artifacts which can be stored and reused later;
3. Templates and services are conformed to standard interfaces. They can be reassembled and reconfigured on demand.

While creating e-business solutions using process templates, we first identify required business services

and IT services. Secondly we search service directory for desired services. We hire them if existed. Otherwise we have to create by our own or request them from other sources. The following step is to define business templates according to business processes, which integrate the services above. Finally, we use these templates to accomplish service composition and create complete e-business solution. The following figure delineates the entire process.

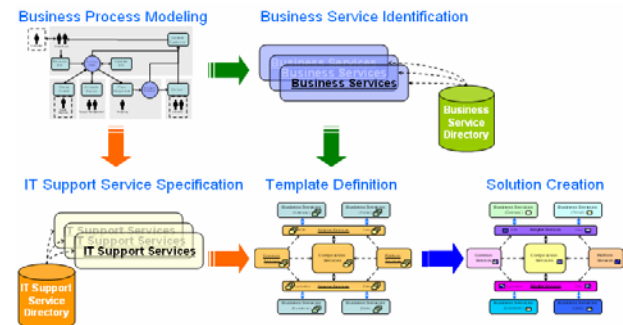


Figure 2. Building e-business solution through template and composition [2, 3]

When business process changes, enterprise only need to develop new service components and define new templates. New application can be easily integrated. Compared to rewriting the whole application, this template-based method reduces cost and makes enterprises more responsive.

3. PROCESS MODELING BASED ON INTELLIGENT WEB SERVICE AND BPEL4WS

More and more enterprises are accepting web-based e-business systems. Web services have been envisioned to be one of the key integration technologies. We have enhanced Web services with intelligence-enabled internal model. Semantic Web Service technologies have not been mature enough to represent knowledge. [8] We therefore incorporate well-developed agent knowledge representation model with Web services internal entities. Such intelligent Web services can contribute to the process composition automation, as well as interaction with partners' systems within on demand business solutions.

BPEL4WS (Business Process Execution Language for Web Service) is an XML-based language used to integrate Web Services to model process. [10] That is, it creates templates to compose Web Services. What created by BPEL4WS could also be a Web Service and described by WSDL. BPEL4WS defined how Web Services involved in the process cooperate and interact with each other to achieve business Goals. To implement on-demand business, we can define several business processes using BPEL4WS, and a composition Web Service will use these templates to invoke other Web Services to accomplish the whole business. Such composition Web Service is implemented using

intelligence and has several business processes modeled with BPEL4WS as its knowledge. It chooses the most appropriate process according to external events (e.g. user request) and internal rules, and executes the selected processes to realize on-demand business. Figure 3 illustrates the architecture of process modeling based on intelligent Web Service and BPEL4WS.

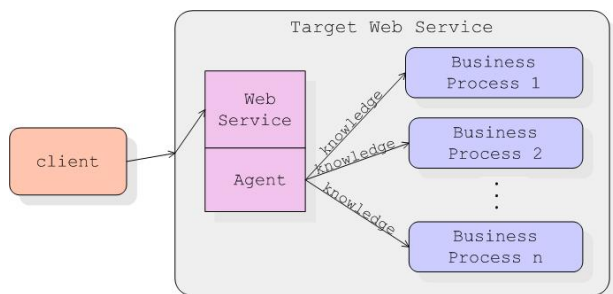


Figure 3. Architecture of process modeling based on intelligent Web Service and BPEL4WS

An end user issues business request through its client software ("client" in above figure), which sends this request to "Target Web Service". "Target Web Service" is implemented with intelligence. It is developed as a software agent with standard Web Service interfaces. With this paradigm, the intelligent Web service can use its own knowledge representation model to incorporate business processes described by BPEL4WS as its knowledge, and then choose the most appropriate one to execute on environment changes and its intelligence. In above figure, from "Business 1" to "Business n" are all represented as agent-applicable knowledge.

4. IMPLEMENTATION ISSUES

For the mentioned case, we use software agent which conforms to FIPA (Foundation for Intelligent Physical Agents) standards to develop intelligent Web service.^[11] The services are implemented as agents and published as Web Services. FIPA agent uses ACL (Agent Communication Language) for communication, which is not compatible with Web Service's communication protocol, SOAP. When using agent to implement Web Service, we must address this discrimination in communication, that is, to make agent understand SOAP. There are two solutions to this problem:

1. Enhance FIPA agent and make it understand SOAP. With this method, agent can communicate with each other using both ACL and SOAP. But it is too difficult since it is almost the same as developing one specific FIPA agent implementation. In addition to difference in communication protocol, which means we have to add parsing and encapsulating of SOAP to agent functions, communication styles also differ. Web Services support synchronous and asynchronous invocation, while FIPA agent can only be called asynchronously. This adds a great deal to the complexity of this method and finally make we give it up.
2. Add a mediator module for FIPA agent, which can

understand both ACL and SOAP. The mediator receives SOAP message from client, extracts request parameters, creates new ACL message, then puts those parameters into new message and finally, sends ACL message to agent. When agent receives ACL message, it uses built-in parameters to perform requested tasks and returns result to the mediator. On receiving ACL-based result, mediator performs another message format translation and sends SOAP-based result back to client. Then one transaction completes. This method is much easier to implement than the previous one. So it is chosen in this paper.

With regard to how to incorporate BPEL4WS-based business process as agent's knowledge, there are also two considerations:

1. Store BPEL4WS files, WSDL files and other assisting files somewhere FIPA agent can access (e.g. file system). When agent activates, it will go to the specified place, read those files and execute processes described in them. On doing so, it is required that agents must understand BPEL4WS and WSDL, as well as perform corresponding operations. Actually, it means adding BPEL4WS engine to agent, which is even more difficult than adding SOAP engine to agent. So it is discarded.
2. Make use of existing BPEL4WS engine and combine it with agent in some way. We choose to deploy BPEL4WS processes on BPEL4WS server and publish as Web Services. Then agent is able to invoke these Web Services with standard interface, as well as accomplish business tasks. When compared to previous one, this method is more convenient and flexible. Therefore, it is the better choice.

5. CASE STUDY: ONLINE AUCTION

For case study, we use the proposed architecture to implement an online auction business process. Customers can issue an auction request on client interface. Then the request is sent to the auction Web service. The service has a few BPEL4WS business templates as its knowledge and knows which one to choose. Then the service invokes the most appropriate BPEL4WS Web service. The selected process is instantiated as an auction session.

Following demonstrate those two business processes involved. One is called "OnlineAuction" and the other "OnlineAuction2". These two processes are identical in the first four steps, described as follows:

1. Customer logins into auction web site, and then enters information about auctioned commodity and criteria used later to evaluate auction result.
2. Auction mediator receives such information and asks buyers to bid.
3. In this case there are two buyers. On receiving auction request, they will bid separately based on the

information customer entered.

4. Auction mediator collects prices from all buyers, calculates their scores, and determines a winner.

In “OnlineAuction”, the step 5 is:

5. Auction mediator returns price and score of winner to customer.

In “OnlineAuction2”, the step 5 is:

6. Auction mediator returns prices and scores of all buyers to customer, and then suggests a winner.

The differences between the two processes are that “OnlineAuction” only returns information about the winner; therefore auction decision is made by the mediator. While in “OnlineAuction2”, mediator returns information about all buyers and suggests a winner, but final decision is still up to the customer.

In this case, we choose a FIPA agent implementation, JADE (Java Agent Development Framework) to develop intelligent Web services.^[9] In order to do so, we also use an open source project called “Wsag”.^[6] It acts as a mediator between the Web Service and JADE agent and makes message format translation. “Wsag” is deployed on Apache Tomcat 4.1.24, while agent is running in JADE 3.1 environment. When publishing BEPL4WS processes as Web Services, we choose an existing BEPL4WS engine, BEPL4WS server 2.0 from collaxa.^[7]

The following figure illustrates architecture of online auction case.

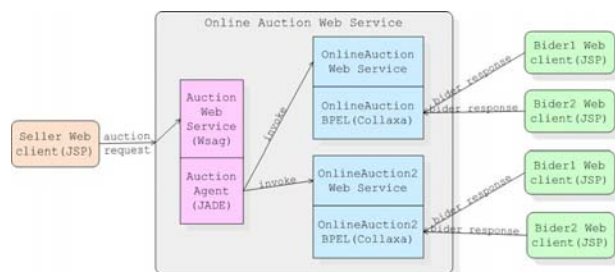


Figure 4. Architecture of online auction case

Following is a complete instance of online auction case. Customer visits auction web site (<http://10.89.103.1:8080/wsag/OnlineAuctionWebClient.jsp>), enters name and description about commodity to be auctioned and specifies weights for buyer's price and reputation. The customer has to choose whether he wants to see bid information of all the buyers or only the winner. In this instance, “Name” and “Description” are set to “Astrology Mythology” and “A Very Interesting Astrology Book”, separately. The weight for “Price” and “Reputation” is 10:1. At last, “Only Return Final Result” is selected, i.e., the customer only wants to see bid information of the winner. This customer request is sent to Auction Web Service, which is implemented with agent and uses “Wsag” as mediator. Auction Agent which accomplishes actual business function is running

on another computer (auction0@SSC036:1099/JADE@<http://10.89.103.36:7778/acc>). When the agent receives request, it knows which process to invoke. If “Only Return Final Result” is selected in this instance, it will invoke OnlineAuction Web Service; otherwise OnlineAuction2 Web Service is called. These two Web Services are implemented with BEPL4WS and deployed on BEPL4WS server running on a third computer (<http://10.89.150.150:9700/collaxa/default/OnlineAuction> and <http://10.89.150.150:9700/collaxa/default/OnlineAuction2>). As aforementioned, these two processes involve human interactions. Buyers can login to web site to participate in these processes (<http://10.89.150.150:9700/OnlineAuctionUI/> and <http://10.89.150.150:9700/OnlineAuction2UI/>). A buyer named “bider1@software” has already login to OnlineAuction web site and sees all process instances waiting for his participation. Then he can examine information of auctioned commodity and criteria, as well as offer price and reputation. After collecting bids from all the buyers, process instance can go on. When calculating scores, process uses following formula:

$$\text{Score} = \text{Price} * \text{PriceWeight} + \text{Reputation} * \text{ReputationWeight} \quad (1)$$

The buyer with the highest score is the winner. Then process will return information of the winner to customer. If customer doesn't choose “Only Return Final Result” at the beginning, process will return information of all the buyers and suggest a winner. Figure 5 shows snapshots for this auction instance.

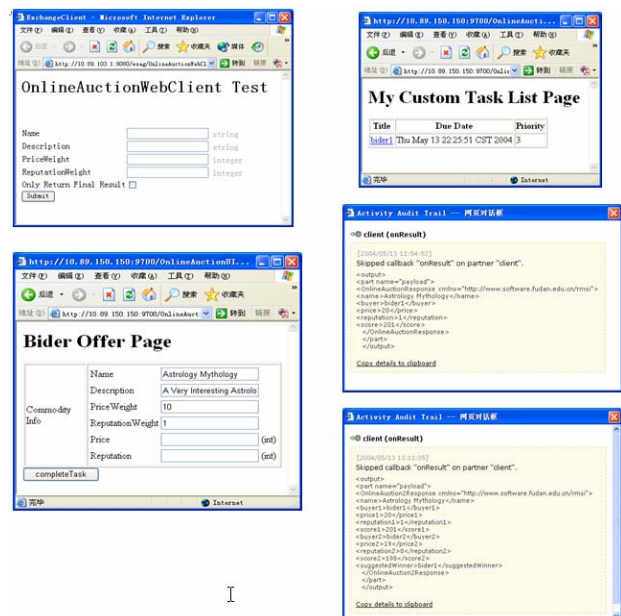


Figure 5. One instance of online auction

In this case, our proposed architecture has been proven feasible. However, Web Service's intelligence is not exploited enough. This needs to be improved.

6. CONCLUSION

This paper proposes that business processes modeled

with BPEL4WS can be made part of intelligence Web service's knowledge to accomplish process modeling, as well as on-demand businesses. Several critical issues have been addressed, including system architecture, how to implement intelligent Web Service with agent-oriented knowledge, and how to represent BPEL4WS processes as knowledge. However, there are several issues that still need improvement, i.e., intelligent Web services themselves don't understand BPEL4WS without existing BPEL4WS engine, and we have not utilized intelligence well to make Web Service intelligent. These problems call for research efforts.

REFERENCES

- [1] Yinsheng Li, Hamada Ghenniwa, Weiming Shena, "Agent-based Web Services Framework and Development Environment"
- [2] Ying Huang, "e-business Solution Templates based on Web Services & Business Processes"
- [3] Ying Huang, "Model-driven integration for service-oriented on demand business solutions"
- [4] Irving Wladawsky-Berger, "Meet the Experts: Irving Wladawsky-Berger on e-business on demand", http://www-900.ibm.com/developerWorks/cn/dmdd/library/techarticles/0302iwb/0302iwb_eng.shtml
- [5] Alfredo Gutierrez, "e-business on demand: A developer's roadmap", http://www-900.ibm.com/developerWorks/cn/ebod/i-ebodov/index_eng.shtml
- [6] Whitestein Technologies, Wsag project, <http://wsai.sourceforge.net/gateway.html>
- [7] Collaxa Company, Collaxa home page, <http://www.collaxa.com/>
- [8] DAML organization, DAML Services home page, <http://www.daml.org/services/owl-s/>
- [9] Tilab, JADE home page, <http://jade.tilab.com/>
- [10] IBM, Microsoft, BEA, "Business Process Execution Language for Web Services Version 1.1 Specification", <ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>
- [11] FIPA organization, FIPA home page, <http://www.fipa.org/>