RC22624 (W0211-016) November 4, 2002
Computer Science

IBM Research Report

Winslow: A Business Process Management System with Web Services

Juhnyoung Lee, Jeaha Yang, Jen-Yao Chung
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Winslow: a Business Process Management System with Web Services

Juhnyoung Lee, Jeaha Yang, Jen-Yao Chung

IBM T. J. Watson Research Center
P. O. Box 218
Yorktown Heights, NY 10598
{jyl, jeaha, jychung}@us.ibm.com

Abstract

Business Process Management (BPM) provides a process-centric paradigm for developing and executing enterprise software applications. It enables the design, analysis, optimization, and automation of business processes. Recently, companies adopt BPM for changing business processes, and implement Business Process Management Systems to improve the effectiveness of their business. Although BPM, especially, in the scope of workflow and transaction management, has been actively studied in the past few years, the design and implementation of Business Process Management Systems on the Service-Oriented Architecture instantiated by Web services needs refreshed attention in a wider scope including process modeling and monitoring. This paper presents a high-level architecture of a Business Process Management System with Web services, and discusses the requirements and the functionalities of its components. It also briefly reviews the current work on developing BPM solutions using Web services centered on an emerging standard called BPEL4WS. Finally, it discusses some of the related research problems whose solution is critical for seamless, effective BPM with Web services.

1. Introduction

Business Process Management (BPM) provides a process-centric paradigm for developing and executing enterprise software applications [16, 18, 23]. A *Business Process Management System* (BPMS) provides a system environment for the design, analysis, optimization, and automation of business processes. When deployed in a distributed environment such as the Internet, it can facilitate collaboration among process participants, i.e., internal and external business partners. BPM provides fundamental advantages over the conventional application-centric paradigm [16]:

- Flexibility in changing the model of the underlying business processes,
- Integration capabilities for even disparate applications,
- Reusability of activity implementation and process models, and
- Scalability of application development and execution.

With these advantages, BPM has a potential to provide business benefits such as reduced costs, improved quality and reduce errors, gaining visibility into the enterprise, automating steps in the process, and balancing workloads. The need to save costs in a tight economy has

sharpened the focus on business-process improvement. Companies employ BPM for changing business processes, and implement Business Process Management Systems to fulfill their needs.

Web services provide technologies for implementing BPM in an efficient, standard way. The promise of Web services is to enable a distributed environment in which any number of applications, or application components, can interoperate seamlessly within an organization or between companies in a platform-neutral, language-neutral fashion [7]. Simply put, in Web service-based BPM, a Web service could represent an activity within a business process, or a complex business process comprising a number of steps.

Coupling of BPM with Web services is also blessing for Web services. As the impact of Web services begins to grow, so will the degree of complexity surrounding this computing paradigm. A Web service could be as simple as running a check on a credit number, and as complex as handling a company procurement process using an auction. To fully leverage the advantages offered by Web services in the delivery of application resources and information, Web service will require the coordination as provided by BPM software in the management of business processes.

Although BPM, especially, in the scope of workflow and transaction management, has been studied in the past few years [1, 4, 14, 16, 20], the design and implementation of Business Process Management Systems using Web services on the Internet needs refreshed attention. This paper highlights a high-level architecture of a BPMS with Web services, and discusses the requirements and the functionalities of its components. It also briefly reviews the current work on developing BPM solutions using Web services centered on an emerging standard called BPEL4WS. Finally, it discusses some of the related research problems whose solution is critical for seamless, effective BPM with Web services. The primary focus of this paper is on general concepts for understanding rather than the underlying technical details. There will be follow-up papers for describing technical details of each of the specific problems. The work presented in this paper is part of an on-going project called Winslow at IBM T. J. Watson Research Center.

The rest of this paper is structured as follows: Section 2 provides a brief introduction to BPM. In Section 3, we present a general architecture of Business Process Management Systems, and describe the requirements and the functionalities of its components. Section 4 summarizes the basic capabilities of Web services and the recent work for BPM with Web services centered on BPEL4WS. It discusses how BPM can be designed and implemented in the Web service architecture. Section 5 discusses some related research problems that are important to the development of effective Web service-based BPM. In Section 6, conclusions are drawn and future work is outlined.

2. Business Process Management

A *process* is a specific ordering of activities across time and space, with commencement and termination, and clearly defined inputs and outputs, in short, an organization for actions. A *business process* refers to a process in which work is organized, coordinated, and focused to produce a valuable product or service [5, 6, 12, 18]. Business processes comprise both

internal and external business partners and drive their collaboration to accomplish shared business goals by enabling highly fluid process networks. *Business Process Management* (BPM) enables the design, analysis, optimization, and automation of business processes. It accomplishes these tasks by separating business logic from application logic, managing relationships among process participants, integrating internal and external process resources, and monitoring process performance.

BPM provides a process-centric computing paradigm for enterprise software applications, and its concept can be better explained in the context of the application system structure evolution [16]. The first-generation enterprise application software was large monolithic pieces of code with a complex internal structure, which combined the infrastructure code for manipulating data, and the pieces of code for implementing the actual logic of the application. The second-generation applications are noted by their use of Database Management Systems (DBMS) which was devised to support the definition and concurrent manipulation of data, and consequently, separate data logic from applications. With DBMS in place, changes to the data schema and access paths, for example, can be done without impacting the related applications.

In a similar way, a BPMS separates business logic from applications by supporting the definition business processes. A BPMS externalizes the definition and execution of the control and data flow of processes, the assignment of people to tasks (or activities), and the invocation of application logic modules. Changes to a process can be done without impacting the related applications. An application becomes a *process-aware application* consisting of a model of the underlying business process (referred to as a *process model* or a *flow model*) and a set of (flow-independent) application logic modules. The abstractions of the elementary pieces of work in a business process are called *activities*; the concrete realizations of these abstractions at process execution time are referred to as *activity implementations*. The application logic modules correspond exactly to these activity implementations in a BPMS. A Process-aware application is illustrated in Figure 1.

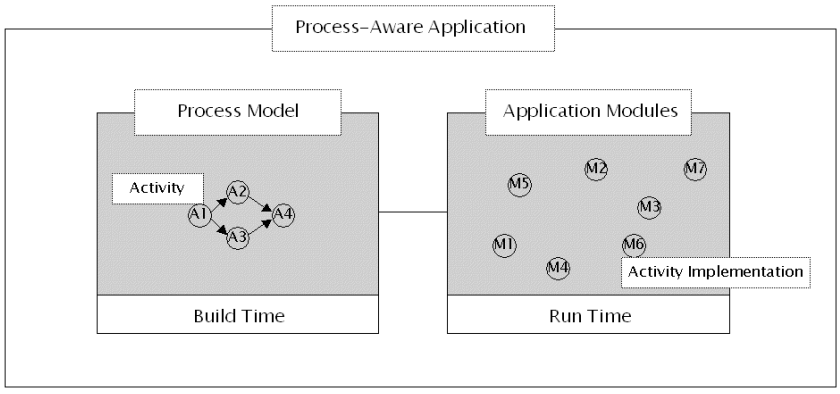

Figure 1 Process-aware application

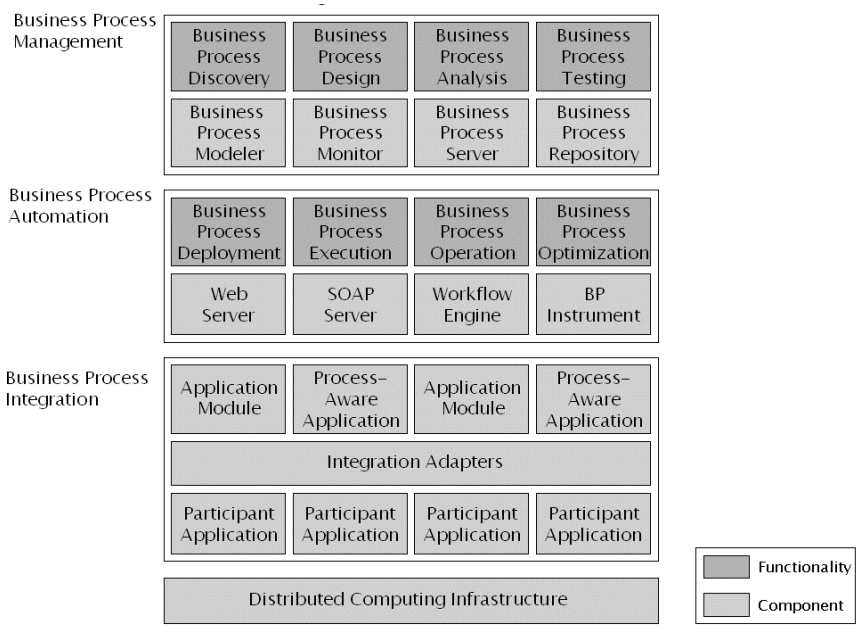

Figure 2 A business process management system architecture

Figure 2 illustrates the architecture of a BPMS. In this section, we briefly explain the logical relationship of three basic building blocks in the architecture, *Business Process Integration* (BPI), *Business Process Automation* (BPA), and *Business Process Management* (BPM). The functionality of each of the components will be described in detail in Section 3. Some of the components that are related to Web services will be discussed in Section 4.

Business Process Integration focuses on the set of activities for creating application modules (as Web services, as will be explained in Section 4) available to business processes or, sometimes, complete process-aware applications (again, as Web services) from participating back-end enterprise applications such as ERP systems. At its core, BPI is simply connection of people and information within a process. To connect application information, APIs (Application Programming Interfaces) of back-end applications, or messaging services are often used. (Then the resulting business objects can be packaged as Web services.) To include human users within the process, often used is user interfaces that have the look and feel of e-mail inboxes. While enterprise application integration (EAI) solutions have typically supported discrete application events, BPI needs to handle extended activity sequences, long-lived processes, compensating transactions, failures and cancellations driven by business requirements [5, 8, 23].

Business Process Automation builds on BPI by providing tools to model, deploy, and operate end-to-end processes. It automates processes by having human activities supported by workflow collaboration. Process automation requires functionalities such as notification services (e.g., e-mails), support for nested processing, and invocation of services to execute processes regardless of modeling language, while persisting state and data between service invocations [18, 23].

Business Process Management builds on BPA by providing tools to model, design, test, and analyze partial or end-to-end processes. It stores process description into a repository and provides a normalized environment within which processes can be designed and managed. At run time, process data that represents state and messages among participating applications or humans is recorded and can be known to BPM software through a process query languages.

3. BPM Requirements and Components

This section focuses on the requirements of BPM, and describes in detail the functionality of each of the components in the BPMS architecture of Figure 2. Some of the components are related to Web services will be discussed in Section 4. In addition to the basic components, the need for a common business process specification language is also explained.

3.1. Common Business Process Specification Language

A business process can be hosted by one of the business partners, and in many cases, the host can be dynamically determined depending on the specific application scenario. As a consequence, the process needs to be specified in a standardized way that can be executed in a number of different process execution environments. Thus, a common language for specifying business processes in a standard way is required to support definition of process models that can be implemented by a variety of process engines.

The emerging BPM industry has been considering multiple alternative paths for modeling of executable business processes. In the Web service domain, Microsoft proposed the adoption of the Pi-Calculus model with XLANG [25], IBM rejuvenated the use of Petri Nets with WSFL (Web Service Flow Language) [15], and BPMI.org unified the two approaches with BPML (Business Process Markup Language) [31]. Along with such parallel efforts, other organizations advocated radically different approaches for business process modeling, such as ebXML BPSS developed by OASIS. Recently, BEA, IBM, and Microsoft jointly released BPEL4WS (Business Process Execution Language for Web Services), a new language for the modeling of executable business processes [33]. In this paper, we will focus on BPEL4WS, which will be explained in detail in Section 4.

3.2. Business Process Modeler

The business process modeler is a tool for modeling business processes, which is the starting point for developing process-aware applications. The ability for the business managers and application developers to design processes without having to write any program is one of the major promises of BPM. The business process modeler tool should enable to efficiently model, simulate, analyze, and validate business processes and software modules. The tool defines a process in terms of business actions and their relations, the organization units performing these business actions, and the business objects that these business actions are working on. The definition of the information is often entered through a graphical user interface with a drag-and-drop capability to make process design as intuitive as possible for the business users. Typically, an animated process graph is used to help determine the

correctness of the process and the correct invocation of the application modules implementing the process activities. Eventually, the model is converted into a physical design in a document format (e.g., in BPEL5WS), which is passed over to the workflow management system for process execution.

Furthermore, a robust process design module should support all process assets (i.e., information and human), sub-processes, parallel processes, creation of business rules, and exception handling among other things. Also, process design should allow process-aware applications comprising application modules linked across multiple, disparate system environments. The tool should understand various aspects of business process integration for achieving optimal performance.

3.3. Business Process Monitor

Business process monitoring should provide business managers with real-time and historical data on the activities of "business-level" services. It should enable a company to detect any changes to the normal operating metrics for any business process or operating parameters in real-time. This data should be interpreted by business analytics software to immediately formulate an optimal response. The result is the ability of the company to more quickly detect important changes and trends and respond to them in a fraction of the time previously required [3, 12]. To collect application data in real-time, the business process monitor gathers information from various transport mechanisms as the data is moved from an application to another. The tool should allow viewing of internal application data through instrumentation and monitoring the business process inside each and every application. Also, the tool should be able to import internal business information from various applications.

The business process monitor presents the business information in an easy-to-access console, often implemented as several dashboards in a company portal environment, to fulfill the needs of different business users. Typically, IT users need to track processes, work items, and business performance measures (modeled in the business process modeler tool in real-time), while taking corrective actions. Upon invocation of a corrective action, the dashboard issues a command to the workflow engine via appropriate APIs. In order to reliably and efficiently operate in a distributed computing environment, system administrators need tools that provide them insights into the status of the systems and networks, and into the status and behavior patterns of their applications. Furthermore, one of the major goals of BPM is to realize continuous process improvement. Thus, a BPMS should provide an administrative console with metrics and reporting capabilities for business users. Through reports on historical data and analyses for trends, companies can take steps toward process optimization.

3.4. Business Process Server

The business process server should provide the ability to view, and share created business process models (and related information) through an easily accessible user interface such as a Web browser via the Internet or company Intranet. The business process server employs a repository that is a secure, central server for cleansing, consolidating, and storing created business process models. It should provide an access control that ensures the intra-company policy or the agreement among business partners over process model distribution and procedure. Some form of version control and change management such as a check-in and

check-out functionality should ensure the integrity of the process models and associated information.

3.5. Workflow Engine

The workflow engine is the run-time component of a BPMS that controls the execution of process instances. It allows the user to start, terminate, suspend, and resume processes. It determines who should perform a particular activity, puts the resulting work item onto the work list of selected user(s), schedules the proper program when a work item is selected and determines what activities come next after one has been completed, and records all these actions in a log.

In addition to the workflow engine, there are several important process operation features associated with process execution. One is run-time modification of business processes. Users need to know that, if anything goes wrong with their process, they can quickly make changes while the process is running and not lose time by starting over from the beginning. In addition, workload balancing is a necessity. In a business environment, both human and software applications will get overloaded, meaning that the BPMS will have to balance the work between all parties in the most efficient manner.

4. Web Services and Business Process Management

An architecture that supports Web services, known as a *Service-Oriented Architecture*, is the latest in a long series of attempts in software engineering that foster the reuse of distributed software components. Unlike its predecessors, Web services make services (implemented as software) available on the Web. Another difference is that the services are described and published in a way that anyone can locate and invoke them. The fundamental of Web services is, in essence, assembling applications over the Web using open interfaces and protocols. Web services are business assets that can be shared, combined, used, and reused by heterogeneous computing resources within an organization or between firms [7].

Technically, a Web service is an XML object comprised of content, application code, process logic, or any combination of these, that can be accessed over any TCP/IP network using the SOAP (Simple Object Access Protocol) standard for integration, the WSDL (Web Services Description Language) standard for self-description, and the UDDI (Universal Description, Discovery, and Integration) standard for registry and discovery within a public or private directory. With the standards and associated tools that enable the specifications, applications developers can:

- Publish and un-publish Web services that are defined using WSDL,
- Find Web services defined using WSDL, and registered and stored in a UDDI registry,
- Create WSDL documents from existing Java classes, servlets and EJB,
- Generate client and server code from a WSDL document, and
- Deploy Web services to a Web application server.

As the impact of Web services begins to grow within many organizations, so will the degree of complexity surrounding this computing paradigm. To fully leverage the advantages

offered by Web services in the delivery of application resources and information requires the same type of coordination as provided by BPM software in the management of business processes.

A business process can be re-defined in terms of Web services: a business process specifies the potential execution order of operations from a collection of Web services, the data shared among the Web services, which business partners are involved and how they are involved in the business process, joint exception handling for collections of Web services, and other issues involving how multiple services and organizations participate [18]. Business processes especially are often long-running transactions comprised of multiple Web services, requiring consistency and reliability of Web services applications.

The Business Process Execution Language for Web Services (BPEL4WS) is an XML-based workflow definition language that allows companies to describe business processes that can both consume and provide Web services [13, 17, 24, 27, 31, 33]. BPEL4WS specification is positioned to become the basis of a Web service standard for composition. Along with complementary specifications, WS-Coordination [36] and WS-Transaction [37], BPEL4WS provides a basis for a business process automation framework. BEA, IBM and Microsoft jointly developed these specifications, replacing old specifications on workflow modeling, WSFL by IBM [15] and XLANG by Microsoft [25].

Along with the BPEL4WS specification, a few tools for this new specification were offered. For instance, IBM provides a workflow engine called BPWS4J on which business processes written in BPEL4WS can be executed, a tool for validating BPEL4WS documents, and an editor for creating and modifying BPEL4WS documents [32]. The BPWS4J engine takes in a BPEL4WS document describing a business process to be executed, WSDL documents describing the component Web services, and a WSDL document describing the interface of the resulting business process to clients. From this input, the engine generates a process made available as a Web service with a SOAP interface. A WSDL file that describes the process's interface may be retrieved at run-time. The BPWS4J engine also supports the invocation, from within the process, of Web services. The workflow engine runs on Web application servers such as Apache Tomcat and IBM WebSphere application server.

The BPWS4J platform also includes an Eclipse [35] plug-in that provides a simple editor for creating and modifying BPEL4WS files. It provide a simple, but versatile process modeler, in that it accommodates both bottom-up and top-down approaches to process design, synchronizes XML source and tree views of business process being created, provides context-sensitive menus that facilitate creation of specification-compliant processes, and validates process against specification requirements during editing session. However, in light of the BPM tool requirements for process modeling, monitoring, and managing described in the previous section, the tools of the BPWS4J framework show apparent limitations in various aspects. Fulfilling the tooling needs of BPEL4WS will be critical for the adoption of BPM with Web services by companies, and present interesting challenges and opportunities for research.

The relationship between a business process and Web services is two-fold: a business process can include Web services as implementation for its activities, and a business process can provide an implementation for a Web services. The former case is illustrated in Figure 3.

A business process acts as a service requester. When the workflow engine, i.e., BPWS4J, arrives at an activity in a process instance, it sends an XML message to the Service Invocation Utility. This message carries all the data needed for the service invocation; the content of the message identifies the Web service and specifies its parameters. Acting as a client to the Web service, the Service Invocation Utility calls the specified Web service through a proxy. Optionally, the Service Invocation Utility can return the result of the Web service as an XML response message to the workflow engine. This message signals the completion of the activity's implementation.

The other case where a business process provides an implementation for a Web services is illustrated in Figure 4. BPEL4WS defines a business process. From this process definition, a developer can derive the definition of a Web service in WSDL that is implemented by this process, including the SOAP binding for this Web service. A developer needs to take the following steps to make a given business process available as a Web service:

- The service provider models a business process using the Business Process Modeler.
- The service provider generates the BPEL4W representation from the model.
- The service provider generates a WSDL file from the BPEL4WS document.
- The WSDL file serves as the Web service interface description for service requesters, and also is used to generate:
  - A server-side skeleton for the service provider,
  - A deployment descriptor for this Java bean, and
  - A client proxy for service requesters.

A Web service request submitted by a service requester is intercepted and transformed into a format understood by the workflow engine, e.g., BPWS4J, which executes an instance of the business process. Note that the execution of the business process can act as a service requester; as it can invoke a number of Web services providing the implementation of activities of the process.

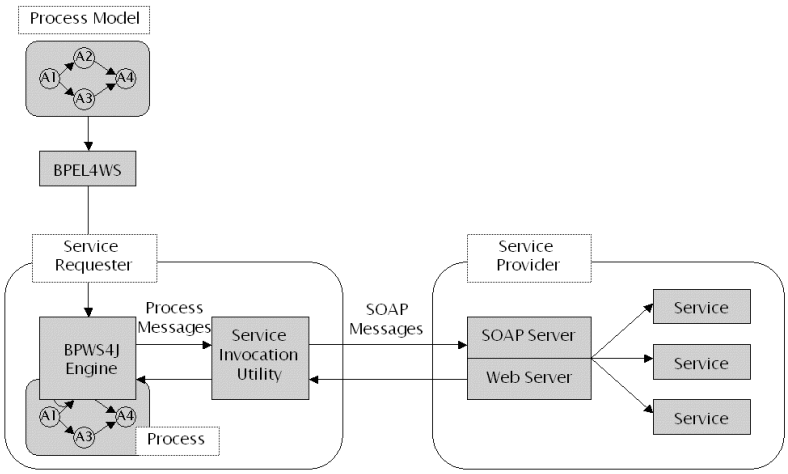

Figure 3 Invoking Web services as activity implementations

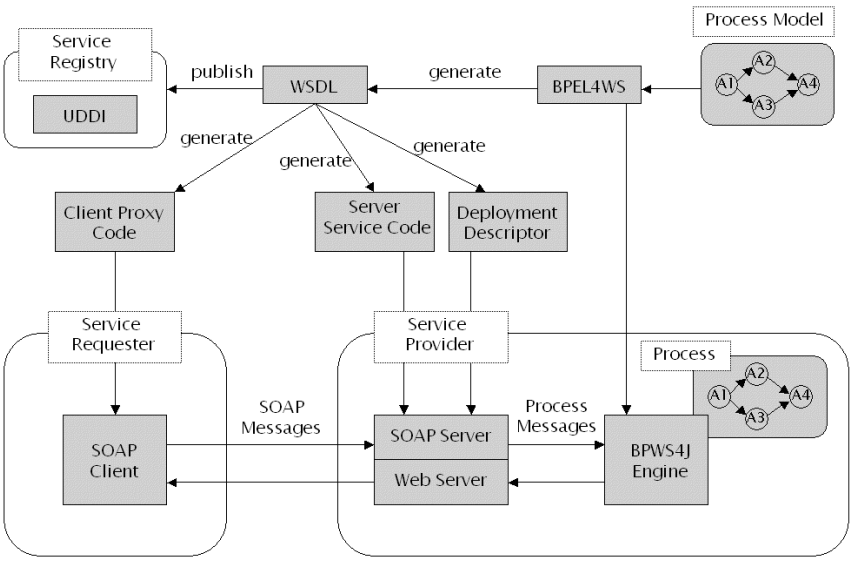

Figure 4 Business process as a Web service

Figure 5 illustrates a view of a BPEL4WS process. BPEL4WS defines a Web service by composing a set of existing services. Like any other Web service, the interface of the composite service is described as a collection of WSDL portTypes. Unlike a traditional software implementation of a WSDL interface, however, each operation of each portType does not map to a separate piece of logic in BPEL4WS. Instead, a single BPEL4WS process implements several portTypes of the service, and the entry-points of specific external users are described in the BPEL4WS document. The entry-points either consume WSDL operations' incoming messages from input-only or input-output operations. In the latter case, the process must also indicate where the output message is generated.

The BPEL4WS process itself is basically a flow-chart like expression of an algorithm. Each step in the process is called an activity. There are a set of primitive activities such as waiting for a message to an operation in the service's interface to be invoked by an external service requester (<receive>), invoking an operation on some Web service (<invoke>), generating the response of an input/output operation (<reply>), waiting for some time (<wait>), and terminating the entire service instance (<terminate>). These primitive activities can be combined into more complex algorithms using structure activities such as <sequence>, <switch>, <while>, <pick>, and <flow>. BPEL4WS also allows recursive combination of the structured activities to express arbitrarily complex algorithms. Details of the BPEL4WS specification and its uses of exemplary business processes can be found in [13, 17, 24, 27, 33].

5. Related Research Areas

With understanding to the basic work on BPM with Web services described in the previous sections, in this section, we briefly review some of the interesting related research problems whose solution is required for seamless, effective BPM in the Service-Oriented Architecture. In particular, this section discusses the following topics that the Winslow project focuses on (in addition to the BPEL4WS tooling discussed in the previous section):

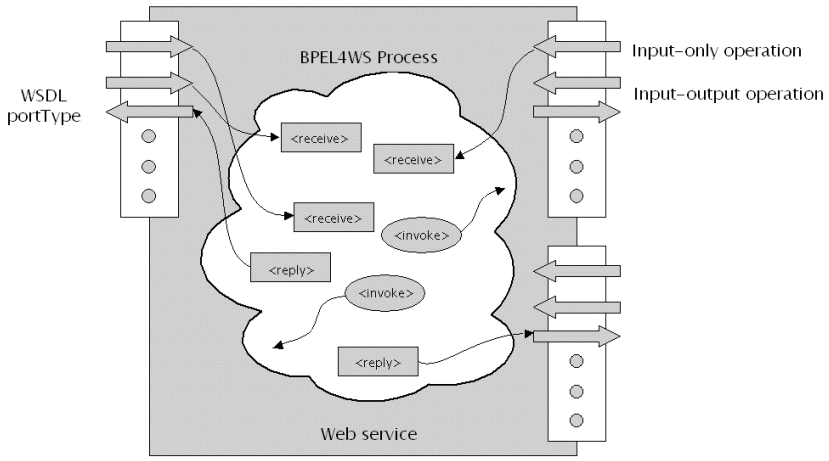

Figure 5 View of a business process implemented as a Web service in BPEL4WS

- Support of transaction management,
- Support of automatic Web service discovery and composition,
- Support of a business process query language, and
- Support of rigorous testing of Web services

Support of transaction management: Support of distributed, long-running business transactions is a difficult problem to solve, and has been actively studied in relation to workflow management [1, 4]. First, the extended duration and communication latency introduced by business tasks performed by involving business partners who run potentially incompatible computing environments introduces complicated requirements. Business transactions must be controlled on a task-by-task basis, with the infrastructure monitoring and managing the results of each individual activity as it relates to the entire process. Another difficulty is that of directly connecting together existing proprietary transaction service implementations. Often, such products simply do not work together "out-of-the-box." Web services can help to build a bridge among those products by providing a common framework through which divergent platforms can be integrated.

The WS-Coordination [36] and WS-Transaction [37] specifications complement BPEL4WS in that they provide a Web service-based approach to improving the dependability of automated, long-running business transactions in an extensible interoperable way. A business process involves a number of Web services working together to provide a common solution. Each of these services needs to be able to coordinate their activities for a successful completion of the process. One method to enforce the coordination of multiple services is to have each Web service share a bit of common information that can be used to link the individual activities to an overall process. The WS-Coordination specification defines a framework through which those services can work from a shared "coordination context." The context contains the information necessary to link the various activities.

WS-Transaction, on the other hand, provides a framework that allows business partners to monitor the success or failure of each individual, coordinated activity. It provides the means for a business partner to monitor an on-going business process and reliably cancel the process in case something goes wrong along the way. If any part of a business process cannot be processed, no part of the process is processed, or, if part of the process has already been processed, the work performed can be compensated for and undone. The WS-Transaction is based on Web services, and thus, transaction support can be made interoperable across vendor-specific transaction management applications. Implementation of WS-Coordination and WS-Transaction specification should be incorporated into the business process automation framework.

Support of automatic Web service discovery and composition: BPEL4WS is basically a language to implement Web services of business processes by composing a set of existing services. It is important to note that service composition for business processes can be enhanced by effective semi-automatic or automatic service discovery [19]. *Automatic Web service discovery* involves automatically locating Web services that provide a particular service and that adhere to requested properties. *Automatic Web service composition* involves the automatic selection, composition, and execution of appropriate Web services to perform some task, given a high-level description of the task's objective.

Recently, there have been studies on computer-interpretable semantic markup of Web services that can be used to specify the information necessary for Web service discovery [9, 10, 19]. Computer-interpretable semantic markup at the service Web sites, and a service registry or (ontology-enhanced) search engine can help automatically locate appropriate services. Another noteworthy work on the automatic Web service discovery and composition is to using rule-based systems [21, 28]. In these studies, a service is represented by a rule, and a rule-based expert system is used to automatically determine whether a desired composite service can be realized using existing services. In case it is possible, the system can construct a plan that, when executed, dynamically instantiates a composite service. Ideally, the business process modeling tools need to provide automatic Web service composition capabilities using automatic Web service discovery or rule-based expert systems, and incorporate them with other basic functionalities such as business process simulation and optimization.

Support of a business process query language: BPEL4WS and its predecessors focus on the modeling and execution of business processes, but it can be ineffective for the management side of business processes. The Business Process Query Language (BPQL) was proposed by BPMI (Business Process Management Initiative: http://www.bpmi.org) organization for the deployment, control, analysis, and optimization of business processes. BPQL is a management interface to a business process management infrastructure that includes a "process execution facility" and a "process deployment facility." The BPQL interface to a process execution facility enables business analysts to query the state and control the execution of process instances managed by the facility. This interface is based on the Simple Object Access Protocol (SOAP). The BPQL interface to a process deployment facility enables business analysts to manage the deployment of process models managed by the facility. This interface is based on the Distributed Authoring and Versioning Protocol. Process models managed by the process deployment facility through the BPQL interface can

be exposed as UDDI services for process registration, advertising, and discovery purposes. BPQL could be useful for business activity monitoring.

Support of rigorous testing of Web services: Testing tools play an important part in the success of BPM with Web services implementation. Quality is more important with Web services than other types of applications, because companies combine the loosely coupled pieces of code that come together to form a business process solution and must work together in a perfect state. It is critical for companies to make the Web services-based BPM applications as reliable as possible, because they involve customers and business partners. There are three main areas of quality testing: functionality (how well the Web service program works), load (how it performs under strain), and regression (whether any code changes result in problems). Ideally, the testing tools for Web services need to be incorporated with the process modeling tools, so that the developers and users can ensure the correct operation of business processes early in the process modeling phase.

6. Concluding Remarks

The need to save costs has sharpened the focus on business-process improvement. Business-technology managers adopt Business Process Management for changing and optimizing business processes, and implement Business Process Management Systems to fulfill their needs. The promise of Web services is to enable a distributed environment in which any number of applications, or application components, can interoperate seamlessly within an organization or between companies in a platform-neutral, language-neutral way. Web services provide useful technologies for implementing BPM in an efficient, standard way.

This paper presented a design of a Business Process Management System in the Service-Oriented Architecture instantiated by Web services. It discussed the logical relationship of three basic building blocks of the BPMS architecture, i.e., Business Process Management, Business Process Automation, and Business Process Integration. It described the functionalities of each of the components, especially in relation with the emerging Web service standard, BPEL4WS (Business Process Execution Language for Web Services) and its development and execution platform. Additionally, the paper briefly reviewed some of the related research problems whose solution is critical for seamless, effective BPM in the Service-Oriented Architecture. The work presented in this paper is part of an on-going project called Winslow at IBM T. J. Watson Research Center. There will be follow-up papers for addressing technical details of specific problems outlined in this paper.

Acknowledgement

We would like thank Henry Chang, Liang-Jie Zhang, and Chae An at IBM T. J. Watson Research Center for their support and help for this work.

References

[1] G. Alonso, D. Agrawal, A. El Abbadi, M. Kamath, R. Gunthor, and C. Mohan, "Advanced Transaction Models in Workflow Contexts," *Proceedings of the 12th International Conference on Data Engineering*, New Orleans, February 1996.

[2] B. Benatallah, M. Dumas, M. C. Fauvet, F. A. Rabhi, and Q. Sheng, "Overview of Some Patterns for Architecting and Managing Composite Web Services," *ACM SIGecom Exchanges*, Vol. 3, No. 3, 9-16, August 2002.
[3] S. P. Bradley, and R. L. Nolan (Editors), *Sense and Respond: Capturing the Value in the Network Era*, Harvard Business School Press, Boston, MA, 1998.
[4] Y. Breitbart, A. Deacon, H. J. Scheck, and G. Weikum, "Merging Application-Centric and Data-Centric Approaches to Support Transaction-Oriented Multi-System Workflows," *SIGMOD Records*, Vol. 22, No. 3, 1993.
[5] E. Brown, "Integrating Business Processes," *Forrester Report*, March 1999.
[6] E. Chabrow, and D. M. Ewalt, "Focus on the Process," *Information Week*, 30-36, August 26, 2002.
[7] D. A. Chappell, and T. Jewell, *Java Web Services*, O'Reilly & Associates, Inc., March 2002.
[8] R. Cutlip, "Business Process Integration with IBM CorssWorlds: Part 1," *IBM developerWorks*, http://www.ibm.com/developerworks/webservices/, May 2002.
[9] D. Fensel, I. Horrocks, F. van Harmelen, D. L. McGuinness, and P. F. Pate, "OIL: An Ontology Infrastructure for the Semantic Web," *IEEE Intelligent Systems*, Vol. 16, No. 2, 2001.
[10] J. Hendler, and D. L. McGuinness, "DARPA Agent Markup Language," *IEEE Intelligent Systems*, Vol. 15, No. 6, 2001.
[11] R. Hull, F. Llirbat, J. Su, G. Dong, B. Kumar, G. Zhou, "Efficient Support for Decision Flows in E-Commerce Applications," *Proceedings of the 2nd International Conference on Telecommunications and Electronic Commerce*, 1999.
[12] J. J. Jeng, S. Buckley, H. Chang, J. Y. Chung, S. Kapoor, J. Kearney, H. Li, and J. Schiefer, "BPSM: An Adaptive Platform for Managing Business Process Solutions," *IBM Technical Report*, IBM Corporation, 2002.
[13] R. Khalaf, "Business Process with BPEL4WS: Learning BPEL4WS, Part 2," *IBM developerWorks*, http://www.ibm.com/developerworks/webservices/, August 2002.
[14] P. Koksal, I. Gingil, and Asuman Dogac, "A Component-Based Workflow System with Dynamic Modifications," *Next Generation Information Technologies and Systems*, 238-255, 1999.
[15] F. Leymann, "Web Services Flow Language," *IBM Corporation*, http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf, 2001.
[16] F. Leymann, and D. Roller, "Workflow-Based Applications," *IBM System Journal*, Vol. 36, No. 1, 1997.
[17] F. Leyman, and D. Roller, "Business Processes in a Web Services World: a Quick Overview of BPEL4WS," *IBM developerWorks*, http://www.ibm.com/developerworks/webservices/, August 2002.
[18] F. Leymann, D. Roller, and M. T. Schmidt, "Web Services and Business Process Management," *IBM Systems Journal*, Vol. 41, No. 2, 2002.
[19] S. A. McIlraith, T. C. Son, and H. Zeng, "Semantic Web Services," *IEEE Intelligent Systems*, March/April 2001.
[20] J. Meng, S. Y. W. Su, H. Lam, and A Hedal, "Achieving Dynamic Inter-Organizational Workflow Management by Integrating Business Processes, Events, and Rules," *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35)*, 2002.
[21] S. R. Ponnekanti, and A. Fox, "SWORD: A Developer Toolkit for Web Service Composition," *Proceedings of the 11th World Wide Web Conference (Web Engineering Track)*, Honolulu, Hawaii, May 7-11, 2002

[22] A. W. Scheer, "ARIS Toolset: A Software Product Is Born," *Information Systems*, Vol. 19, No. 8, 607-624, 1994.
[23] H. Smith, and P. Fingar, "Business Process Management Systems Environmental Policy," *Internet World*, http://www.internetworld.com, May 2002.
[24] J. Snell, "Automating Business Processes and Transactions in Web Services: An Introduction to BPELWS, WS-Coordination, and WS-Transaction," *IBM developerWorks*, http://www.ibm.com/developerworks/webservices/, August 2002.
[25] S. Thatte, "XLANG – Web Services for Business Process Design," *Microsoft Corporation*, http://www.gotdotnet.com/team/xml_wsspecs/xlang-c, 2001.
[26] J. Thoth, "BPM Utilizing the Exception Modeling Pattern," *White Paper*, Holosofx Corporation, http://www.holosofx.com/, 2002
[27] S. Weerawarana, and F. Curbera, "Business Process with BPEL4WS: Understanding BPEL4WS, Part 1," *IBM developerWorks*, http://www.ibm.com/developerworks/webservices/, August 2002.
[28] L. Zeng, D. Flaxer, H. Chang, J. J. Jeng, "PLMflow – Dynamic Business Process Composition and Execution by Rule Inference," *IBM Technical Report*, IBM Research Division, 2002.
[29] L. Zhang, H, Chang, and T. Chao, " Web Services Relationships Binding for Dynamic e-Business Integration," *IBM Technical Report*, IBM Research Division, 2002.
[30] "BPM 2002: Market Milestone Report, Assessment of Q-Link Technologies" *A Delphi Group White Paper*, http://www.qlinktech.com/, 2002.
[31] "BPML and BPEL4WS: A Convergence Path Toward a Standard BPM Stack," The Business Process Management Initiative, http://www.bpmi.org, August 15, 2002.
[32] "BPWS4J," IBM Corporation, http://alphaworks.ibm.com/tech/bpws4j, August 2002.
[33] "Business Process Execution Language for Web Services, Version 1.0," BEA Systems, IBM Corporation, and Microsoft Corporation, Inc., http://www.ibm.com/developerworks/library/ws-bpel/, July 2002.
[34] "Business Process Management and Its Value to the Enterprise," *White Paper*, HandySoft Corporation, http://www.handysoft.com/, 2002.
[35] "Eclipse Platform Technical Overview," *White Paper*, Object Technology International, Inc., http://www.eclipse.org/, July 2001.
[36] "Web Services Coordination (WS-Coordination)," BEA Systems, IBM Corporation, and Microsoft Corporation, Inc., http://www.ibm.com/developerworks/library/ws-coor/, August 2002.
[37] "Web Services Transaction (WS-Transaction)," BEA Systems, IBM Corporation, and Microsoft Corporation, Inc., http://www.ibm.com/developerworks/library/ws-transpec/, August 2002.