SERVICE ORIENTED ARCHITECTURES FOR BUSINESS PROCESS MANAGEMENT SYSTEMS

ABSTRACT

The integration of research in the area of Service Oriented Architectures (SOA) with current methodologies and tools of Business Processes Management Systems (BPMS) will bring significant contributions to the objectives of dynamic process management in modern enterprises. However, at the current state of development of both BPMS and SOA, this dynamism is limited by numerous factors. This article aims to provide a brief survey of on-going research in the application of Service Oriented Architectures in Business Process Management, examining its usage, its relation with other technologies and related open issues. In particular we take a specific view of the problem, namely that of the software engineer that is asked to design, develop and implement service architectures under current business process management systems. We proceed presenting a simple case study that captures some of the proposed methodologies and tools and we use such case study to highlight a number of problems related to the implementation of such architectures in real world situations.

KEYWORDS

Service Oriented Architectures, Business Processes Management Systems, Web Services, e-Commerce

1. INTRODUCTION

Change is the only constant in today's business. Enterprises do not make contracts/agreements for long-term periods. On the contrary alliances are made for short-period strategies that can last some days and even less. In this dynamical environment, enterprises cannot survive without well-organized and adaptive business processes. To face the challenge, current business processes, consisting of numerous habits, practices, disjoint data models, application logic, workflows and many other point solutions repeated a hundred times in a hundred places, need to be rationalized. What is sought out by corporate enterprises is not a new "silver bullet" system to replace existing "legacy" systems, nor a new "business process layer" in an already complex IT stack. Rather, companies need the capability to recast all business processes into a standard form that is open to manipulation by the familiar tools and skills already in place.

To achieve such fundamental shift from *process reengineering* to *dynamic process management*, companies are in the process of investigating and experimenting new systems such as Business Process Management System (BPMS) (see, for example http://www.bpmg.org/) as well as emerging standardized architectures like Service-Oriented Architectures (SOA), a new methodology for building distributed applications where elementary business processes, exposed as services, can be published, discovered and bound together to create more complex valued-added business processes [Papazoglou and Dubrey, 2004].

This article aims at providing a brief survey of on-going research in the application of Service Oriented Architectures in business process management, examining its usage, its relation with other technologies and related open issues. In particular we take a specific view of the problem, namely a developer point of view on service architectures under current business process management systems. We proceed presenting a simple case study that captures some of the proposed methodologies, tools and we use such case study to highlight a number of problems related to the implementation of such architectures in real world situations.

2. BUSINESS PROCESS MANAGEMENT AND SERVICE ORIENTED ARCHITECTURES

2.1 Overview

A business process is viewed as any set of activities performed by a "company" that is initiated by an event, transforms information, materials and/or business commitments, and produces an output. Value chains and large-scale business processes produce outputs that are valued by customers. Other processes generate outputs that are valued by other processes (see references in http://www.bptrends.com). Following the above definition of a business process, supply chain can be viewed as one representation of a business process. In this paper we will use both definitions with the same meaning.

Accordingly to the view presented in the introduction, current business processes need to quickly react on customer needs and market conditions. There are several main directions in business process management research, among which: (1) Industrial solutions for business activities support; (2) Business process modeling; (3) SOA-based business processes management;

Industrial solutions for business activities support

Systems that support supply chain management activities in the industry are still relying on tightly-coupled connections. Because of this they cannot produce fast responses to the changes in the real world. Furthermore, because of the evolutionary nature of the progress in the industry now these systems can handle only part of the company activities. The current role of supply chain management systems can be categorized in a long list of industrial solutions for specific business activities [following Hugos, 2003]:

- Enterprise Resource Planning (ERP)
- Procurement Systems
- Advanced Planning and Scheduling
- Transportation Planning Systems
- Demand Planning
- Customer Relationship Management (CRM) and Sales Force Automation (SFA)
- Inventory Management Systems
- Manufacturing Execution Systems (MES)
- Transportation Scheduling Systems
- Warehouse Management Systems (WMS)
- Workflow Automation Systems

Obviously, there are several attempts to combine all of them in one single system, but such attempts face the natural intricacies of complex systems. Among these: multi-party data transformations, asynchronous interactions, exception management, operations management and others. To the best of our knowledge none of these attempts has succeeded to build a solid and scalable system at the moment.

Business process modeling

Ideally, it should be possible to create a logical model of a business process with the specified details accuracy. However, the modeling environment should be simple enough to be understandable for business people and do not require any special knowledge. At present, there are several languages for business process modeling that are already de-facto standard in this area - ebXML (http://ebxml.org) and BPEL (http://www-106.ibm.com/developerworks/library/ws-bpel/) for example - but in their current version, they are too technically complex to be used directly by the final users, i.e. business managers. To overcome such difficulties, software vendors are evolving such languages towards higher level modeling languages connected to graphical tools with further automatic BPEL or ebXML code generation.
Some of the main players in this field, are:

- Sybase Power Designer (http://www.sybase.com/products/developmentintegration/powerdesigner/): allows non-technical business processes description with further BPEL or ebXML generation. It is based on the UML activity diagram with some simplification to be more understandable for business users.
- Oracle BPEL Editor (http://www.oracle.com/technology/bpel/): allows direct business processes modeling with BPEL. Requires detailed knowledge of BPEL.
- Enhydra Shark (http://shark.enhydra.org/): uses XPDL as a language for describing business processes.
- IBM WebSphere (Integration Edition) (http://www-106.ibm.com/developerworks/websphere/): allows modeling business processes with BPEL.

To model any business process, it is necessary first to obtain its formal description, i.e. it is necessary to find a suitable model description language. As the most accepted in the software industry, Unified Model Language (UML) has been also applied for business processes description [Eriksson and Penker, 2000]. UML gives really good visual representation, but it cannot guarantee desired level of description formalism needed for further actions (like verification, quality of services etc.). Application of more formal methods to business processes includes business process model translation to the logical representation (for example, process algebra) and further verification [Koehler, et al, 2002]. Another similar approach includes business processes verification using Petri nets [van der Aalst, 2003].

Current business process modeling environments (like Rational Rose) include graphical tools which allow user to represent activities inside business process using a palette of specified graphical symbols. It follows that users deal only with a static view of the process. More advanced tools allow doing some calculation of the process times or even basic process analysis. However, most of these tools are not able to conduct "what if" analysis to show the dynamic change of business processes and evaluate the effects of casual events and random behavior of resources. This is supported only by using simulation models of business processes [Irani, et al, 2001].

SOA-based business processes management;

Business process management leveraging standardized Service-Oriented Architectures (SOA) is currently evolving solid solutions to achieve business processes flexibility [Leymann, et al, 2002]. SOA is a logical way of designing a software system to provide services to either end-user applications or to other services distributed in a network, via published and discoverable interfaces. *Service descriptions* are used to advertise the service capabilities, interface, behaviour, and quality. Publication of such information describing available services (in a service registry) provides the necessary means for the discovery, selection, binding, and composition of services. In particular, the *service interface description* publishes the service signature while the *service capability description* states the conceptual purpose and expected results of the service. The expected behaviour of a service during its execution is described by its *service behaviour description* (e.g., as a workflow process). Finally, the *Quality of Service* (*QoS) description* publishes important functional and non-functional service quality attributes, such as service metering and cost, performance metrics (response time, for instance), security attributes, transactional integrity, reliability, scalability, and availability. The conventional SOA characteristics outlined previously are described in the basic services layer of the extended SOA proposed by [Papazoglou and Dubrey, 2004].

Several promising trends have been gaining momentum and recognition within the last decade in this new field, in particular self-organized networking, service provisioning and automatic service composition:

i. *self-organized networking* deals with dynamism of the real-world, elaborating methods and techniques required to analyze, utilize and manage open environment constructions.
ii. the central idea of *service provisioning* originates from the business environment. It was first utilized in software development through the component development paradigm, as a logical extension of the OOP approach. It evolved in the ASP (Applications Service Provider) software model: the ASP is a third party entity that deploys, hosts and manages access to a packaged application and delivers software-based services and solutions to customers across a wide area network from a central data centre. Although the ASP model introduced the concept of software-as-a-service first, it suffered from several inherent limitations such as the inability to develop highly interactive applications and inability to provide complete customisable applications. Today we are in the midst of another significant development in the evolution of software-as-a-service: with the growth of the service oriented approach it became possible to overcome these limitations – service-based software modeled and tailored to the business world and capable of following its dynamic evolution.

iii. *services composition* techniques have been studied for a long time – new languages and methods are being proposed by different institutions. However, additional effort is needed in the analysis of their capabilities and limitations [Wohed, et al, 2003]. Moreover, services composition is of interest not only as a separate issue, but also as a mean of automatic acquisition/creation of new services. The high-level abstraction of this problem brings us the vision of the supply chain as a service itself – i.e. with an automatic services composition we could obtain a viable route for automatic supply chain generation.

At present, the paradigm of Service Oriented Computing is not yet completely mature for application at industrial level. Some foundation concepts appear in [Gold, et al, 2004] and present service-oriented approach to the software design for some steps of the development. However, a comprehensive methodological approach is still missing. Preliminary work is in progress [Papazoglou, 2004]: this work proposes a complete methodology for service life-cycle and presents some guidelines for its application within the web-services applications.

2.2 State-of-the-art analysis

We believe that the integration of research results within SOA with current tools and methodologies of Business Processes Management Systems will bring significant contribution to the objectives of dynamic process management in modern enterprises. However, at the current state of development of BPMS and SOA, this dynamismis limited by numerous factors, namely:

- Automatic services discovery is limited, among other things, by computers inability to understand human language semantics. According to the WSDL specification draft, services description can exists in a form of human readable and/or machine processable text. On one hand it is impossible to classify services to the application areas based on their informal description. On the other hand, formal description also does not give us enough information. For example risk analysis for banking and insurance systems can be hardly distinguished automatically.
- Dynamic services' binding is limited by services inability to discover partner services' security metrics. This problem belongs to reliability, trust and Quality of Services (QoS) issues. For example, to bind service it is necessary to verify that service conform the process requirements/expectations. This is a typical verification problem. Another example raises trust issues: is the service reliable enough for processing clients' credit cards information?
- Dynamic services execution is limited by services inability to automatically process business objects from different domains. This problem belongs to the data representation and matching domain.

Most of these limitations depend on the description formalization problem.

On the other hand business processes modeling area itself has not evolved yet towards a formal description (description representation problem) for even static models (with bindings before execution). The same problem also limits formal methods application for processes verification. Current methodological approaches for Service Oriented Architectures have to clearly answer to the same kind of questions G. Booch [Booch, 1995] has analyzed and provided solution for the object-oriented methodology: How do I evolve my organization to service-oriented practices? What artifacts should I manage to retain control? How should I organize my code ? How do I measure the quality of the software being produced? Can service-orientation help me, help my customers better articulate what they really want?

The required answers to these kind of questions can be obtained only with the successful application of the methodology in the real-world projects.

In order to begin to answer, at least partially, these questions, it is useful to use a prototype system based on SOA that addresses some real-world problems. Such system can be used both as an experimental platform for testing services composition and management techniques and as a case-study example. This is the approach that we have initiated in our research. In the next section an overview of the selected case-study, its architecture and implementation is provided. In the final section we will discuss some preliminary results and more specifically address some open issues related to the implementation of such architectures in real environments.

3. CASE STUDY: VIRTUAL SHOP SERVICE

The aim of the present case study is to evaluate recent technologies in the design, implementation and deployment of SOA architectures for Business Process Management. We have focused on a virtual shop aggregated service. Briefly, the aims of any such a shop can be summarized as follows:

- present the products to the customer (with some constrains, like a 24x7 format);
- support possibility to try products before buying (evaluation versions);
- support order placing process;
- support the order processing process (like declining or accepting purchase);
- deliver the requested products;

Our simple example focuses on a simple eCommerce case-study: the sale of a particular kind of intellectual property, namely software. This simplifies the process of goods delivery. Obviously, there are already many all-in-one systems (like Yahoo! Small Business, ShareIt!, Handango and others) that support all the activities described above. However, from the practical point of view their fees are not always appropriate. For example, all bank transaction can be handled by software vender's bank directly, eliminating additional fees. On the other hand more complex operations like online transactions with credit cards need to be handled by specific services.

3.1 Description and architecture

Scenario

Motivating scenario concerns buying software in online store. First we will examine the customer's view on the overall process: customer chose the desired product and proceed to check out; he/she fill in the form and either purchase online or chose to purchase offline; in case of online payment the product will be delivered in minutes; in case of offline payments the customer receives invoice either by fax or/and by email and goes to the bank; after purchasing the product customer receives payment confirmation and desired product. Please, note that , from his perspective, customer is dealing with a single system.

On the other hand, the main goal of the business owner is to minimize overhead expenses and increase profit. There are already all-in-one shops that can sell his/her products with all possible payment methods. However, they charge important amount for each transaction. To reduce these costs a business owner can decide to handle all wire money transfers directly by his bank, while handling all other means of payments (like credit cards) through existing shop. So, from his point of view there are two system to support: (1) one-stop online shop and (2) a personalized system for wire transfers handling.

In our present example we developed an aggregated service that supports both system. The suggested architecture allows minimizing costs of software delivery to customers. In Figure 1 to 3 we summarize the scenario in appropriate use-case diagram with actors and main operations involved.

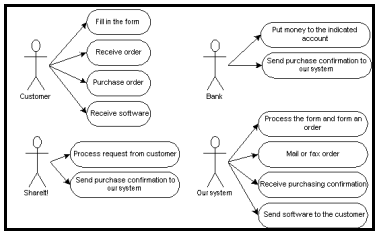

Figure 1: Actors and their activities

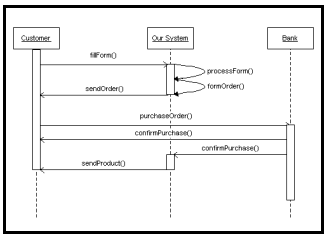

Figure 2: Communications with bank

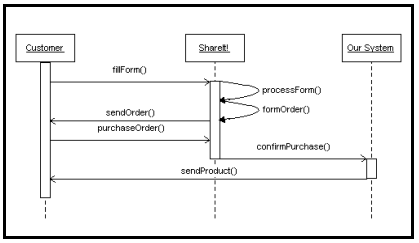

Figure 3: Communications with external service

3.2 Implementation: development platforms evaluation

During implementation we have evaluated the following platforms:

- Sun Microsystems JWSDP v1.3 + Tomcat 5.0
- IBM BPWS4J v2.0 + Apache Axis 1.1 + Apache Tomcat 4.0
- Oracle BPEL Process Manager v2.0 (former Collaxa BPEL Server) + WebLogic 8.1

In this section, we discuss benefits and limitations aroused in the implementation of our case study for each platform.

Sun Microsystems JWSDP v1.3 + Tomcat 5.0

JWSDP from Sun Microsystems is a collection of scripts for generation and deployment of web-services. The approach of having separate script for special domain applications is appealing and has already proved its advantages in Linux-based systems. A clear advantage of such framework is that all deployment procedures can be integrated into Jakarta Ant script. However, in the current version of JWSDP such script collection is intricate to use, lacking a specific development supporting environment. Each script has a large number of parameters and its output have complex relationships with other scripts. The developer is involved in all small details of web-services deployment procedure, so he/she has to be an expert in all different technologies connected to web-services. More problems arise when the developer needs to customized the application during deployment.

IBM BPWS4J v2.0 + Apache Axis 1.1 + Apache Tomcat 4.0

The development environment based on IBM BPWS4J has shown a better performance and stability for the services we needed to support in our case study. The most significant advantage of this platform is that it is light-weight, simple to deal with and can be customized without major difficulties. However, the original BPWS4J package is missing some capabilities that are essential for any medium size project: namely,

- it cannot be integrated into popular development environments based either on Apache Ant or makefiles;
- the deployment of the business processes is limited to the web-browser

To eliminate these shortcoming a BPWS4J Deployment Pack has been developed by IBM: it adds to the original BPWS4J package the ability to integrate business processes deployment within the project, in particular, deployment and undeployment of the business processes from makefile deployment descriptor, Jakarta Ant build tool or just command-line scripts as well as some business process monitoring capabilities.

Customization after deployment is possible, but not well supported and documented in current version. After development and deployment of the system prototype, we tried to extend the system to support different invoices types. This involves passing arrays to the services during invocation. Such functionality is already described in the specification but current BPWS4J implementation is still not supporting it, even in more recent version (BPWS4J 2.1).

An important limitation is that all WSDL and BPEL files need to be hard coded by hand. As with the previous platform, the lack of an integrated environment dedicated to web-service development and deployment limits the overall platform usability, particularly in debugging phases.

Oracle BPEL Process Manager v2.0 (former Collaxa BPEL Server) + WebLogic 8.1

Other alternative platform for service-based systems is Oracle BPEL Process Manager. It comes as an extension to the most popular EJB-containers – JBoss, WebLogic and SunOne. It has special IDE based on the Eclipse environment, for designing and managing processes based on BPEL with complete support of the Oracle BPEL Process Manager. It is therefore possible to design, implement, verify, deploy and test your processes within the same development environment. To the best of our knowledge this system is the most stable and ready for use in production at the moment.

More implementation details and source code for this case study and for the three analyzed platform can be found on the project web-page: http://www.dit.unitn.it/~chall/ .

3.3 Implementation: a path for improvements

As an example of a viable path to improve current development and deployment strategies in current version of development platforms, we focus on the problems short-listed in § 2.2, namely: automatic services discovery, dynamic services binding, dynamic services execution.

In order to gain insight and experience on the various issues involved in the problem, we have sketched and investigated a semi-automated solution for the specific example. To support service discovery, binding and execution we have analyzed and applied the following scheme:

1. Define general process flow in BPEL (for example, using BPEL Designer);
2. Map manually all the necessary services in the BPEL file;
3. Verify both formally and operationally the compiled business process with tools provided by the chosen development environment;
4. Extend BPEL in order to support each service that can have alternative mapping (for more details o the proposed extensions see Figure 4 and text below);
5. Deploy resulting BPEL file within the preprocessor.

```
<invoke inputVariable="orderData" outputVariable="isFormValid" applyType="notify">
  <description operationName="form validation, type validation">
    <input type="tns:orderDataMap">value/type map</input>
    <output type="boolean">true/false</output>
  </description>
  <testsuite>
    <testcase>
      <input value="tns:testOrder1Data"/>
      <output value="tns:testOrder1Result"/>
    </testcase>
    <testcase>
      <input value="tns:testOrder2Data"/>
    </testcase>
  </testsuite>
  <preferred switch="costValue>5">
    <invoke name="verifyForm" partnerLink="verifier" portType="tns:formVerifier" operation="verify"/>
  </preferred>
</invoke>
```

Figure 4: An extended BPEL process flow for PurchaseOrder process

In the short example presented in Figure 4, a few extensions to the BPEL language have been proposed and used, in particular:

- the invoke operation have an **applyType** attribute for indication what preprocessor should be chosen.
- the **Description** tag includes both formal and informal description of the input and output data for the service. In particular, informal description can be used to find services in the UDDI registry, while formal description to automatically find mappings for services published by different providers.
- the **Testsuite** tag indicates tests that should be used for the new service verification. Output value can be either specified or obtained from the service used at the moment of verification.
- the **Preferred** tag indicates the service that should be used if there is no other alternative found.
- the **Switch** attribute specifies the condition when preferred service should be used even if new service was found.

We perceive the suggested approach, to the specific case study, as an evolutionary scheme towards a viable solution: it is based on current platforms, it is used to test current architectures, and proposes optimized scheme to be included in new releases of development environments for web services.

4. CONCLUSION AND FUTURE PLANS

Based on the analysis of both the current state of the research and on the problem encountered in the described case study we are at present focusing on several promising direction in our research, namely:

- Service-Oriented Programming Methodology: we want to investigate the introduction of rules and best practices of complete cycle of software development. By doing so, we would like to outline similarity and differences with existing approaches like Object-Oriented and Component-Based Software Engineering.
- Services Binding: the process of services binding is a complex process to automate. Two directions are available: (1) static services binding on the phase of supply chain modeling; although this is the simplest situation possible, it rejects any future model changes by definition; so, evolution of the process is not manageable. (2) dynamic services binding during process execution: this is the hardest route; it includes search for the next service in chain only after the previous step was passed; obviously it results in a completely manageable business process.
- Services Life-Cycle Management: services manageability is defined as the functionality required for discovering the existence, availability, performance, health, patterns of usage, extensibility, as well as the control and configuration, life-cycle support and maintenance of a service or business process. Management of services can be achieved in terms of detailed application performance statistics that support the assessment of the application effectiveness, permitting complete visibility into individual business processes, and delivering application status notifications when a particular activity is completed or when a decision condition is reached. The field of service management has practically received very little attention so far and no deep investigation of problems relating to service management exists. This is even more acute when there is a need to consider how rich behavioral description of services (including business rules) affects service compositions.

REFERENCES

Booch, G, 1995. *Object Solutions : Managing the Object-Oriented Project*. Addison Wesley Professional

Eriksson, H. and Penker, M., 2000. *Business Modeling with UML: Business Patterns at Work*. John Wiley & Son, Inc, New Jersey, USA

Gold, N., et al, 2004. Understanding Service-Oriented Software. *In IEEE Software*, No. 3, pp 253-265.

Hugos, M, 2003. *Essentials of the Supply Chain Management*. John Wiley & Sons, Inc, New Jersey, USA.

Irani, Z., et al, 2001. Business Process Re-engineering: A Modelling Perspective, *In International Journal of Flexible Manufacturing Systems*, Vol. 13 No. 2, pp. 99-104

Leymann, F., et al, 2002. Web services and business process management. *In IBM Systems Journal*, Vol. 41, No. 2, pp 253-265.

Koehler J., et al, 2002. From Business Process Model to Consistent Implementation: A Case for Formal Veri?cation Methods, *Proceedings of 6th International Conference of Enterprise Distributed Object Computing*, 17–20 September 2002, Lausanne, Switzerland.

Papazoglou, M. P. and Dubray, J., 2004, A Survey of Web service technologies. *Technical Report DIT-04-058*, Department of Information and Communication Technology, University of Trento, http://eprints.biblio.unitn.it/archive/00000586/

Papazoglou M. and van den Heuvel W. "A Business Centric Methodology for Developing WebService Based Applications", *private communication*, to be published.

Wohed P., et al, 2003 Analysis of Web Services Composition Languages: The Case of BPEL4WS, *In Lecture Notes in Computer Science*, Vol. 2813, pp 200-215

Wohed P., W.M.P. van der Aalst, Dumas M., H.M. ter Hofstede A., "Analysis of Web Services Composition Languages: The Case of BPEL4WS", In I.Y. Song, S.W. Liddle, T.W. Ling, and P. Scheuermann, editors, *22nd International Conference on Conceptual Modeling (ER 2003)*, vol. 2813 of Lecture Notes in Computer Science, pages 200-215. Springer-Verlag, Berlin, 2003.