October 2011

Working of SOA with BPM

Mrs. Aarti M. Karande

V.J.T.I. Mumbai, India, aartimkarande@rediff.com

Mr. B.B. Meshram

V.J.T.I. Mumbai, India, bbmeshram@yjt.ac.in

Follow this and additional works at: <https://www.interscience.in/ijcsi>

Part of the [Computer Engineering Commons](#), [Information Security Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

Karande, Mrs. Aarti M. and Meshram, Mr. B.B. (2011) "Working of SOA with BPM," *International Journal of Computer Science and Informatics*: Vol. 1 : Iss. 2 , Article 2.

DOI: 10.47893/IJCSI.2011.1015

Available at: <https://www.interscience.in/ijcsi/vol1/iss2/2>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer Science and Informatics by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

Working of SOA with BPM

Mrs. Aarti M. Karande
V.J.T.I. Mumbai, India
Email: aartimkarande@rediff.com

Mr. B.B. Meshram
V.J.T.I. Mumbai, India
bbmeshram@vjti.ac.in

Abstract

Business processes which are the set of logical tasks related to fulfil some greater purpose. These tasks must be performed in sequence according to the business rules. SOA reuses services to automate a business process using a standard interface and message structure. BPM, based on workflow technology, enterprise application integration and XML technologies, manages the enterprise from the perspective of business processes. In software development there is a gap between the implementers and the users/customers in their understanding of the system. SOA is the driver for minimizing the gap between business analysis and IT development. The idea of Business Processes based on web services stem from the thought that a Business Process can be composed from one or more web services. Service orchestration should be flexible and adaptable to meet changing Business Process requirements. Separating the business process logic from the service components promotes flexibility. Choreography languages are WS-CDL, Let's Dance, and BPEL4Chor. The purpose of using several choreographies simultaneously is to establish a cross-enterprise organized way of collaborating with services.

Keywords- SOA, web services, BPEL, Choreography, WSBPEL, XML

1 INTRODUCTION

Business processes are basically a set of logical tasks which are related to fulfil some greater purpose. These tasks must be performed in sequence according to the business rules. A Business Process can be long-lived or short-lived depending on its function [1].

The general idea in SOA is to expose services for others to use. This is achieved by applying loose coupling and platform neutral interfaces which allow the configuration of the services to be flexible and dynamic. A service is expected to be coarse grained and reusable which makes it ideal for use with BPM. BPM needs services to fulfil the process flow and in the case of combined SOA and BPM these services are conveniently exposed by the SOA. However SOA and BPM do not only exist in symbiosis, SOA can exist without BPM and vice versa. SOA is the driver for minimizing the gap between business analysis and IT development.[2]

A. Service Oriented Architecture

With technology of Service-oriented Architecture, It is easy for developers to integrate the old IT applications with new IT applications by using Service-oriented Architecture.

Figure 1. Service-oriented Architecture Model

As shown in Figure 1, Service-oriented Architecture includes three roles: Service Provider, Service Requester and Service Broker. Service Provider publishes its services and makes the services available to the users who need them. Service Requester is consumer of the services. It can be an application client who needs the services provided by Service Provider. Service Broker is a service registry, which is a kind of agency. It gives a way for Service Provider and Service Requester to connect with each other. Service Provider publishes the web service to Service Broker, that is, to tell Service Broker there is such a web service, where the web service can be found and how to access the web service. So Service Broker will get a collection of web services. Service Broker publishes the web services information to describe the services and the exact location, which can be considered as a menu, so that Service Requester is able to gain the required service by the menu. With the information, Service Requester will be able to access or invoke the web services.

B. SOA reference architecture

SOA reference architecture includes various abilities that need to be considered during implementation of SOA in enterprises, whatever products and services the enterprise chooses. The spine of reference architecture is the joint level Enterprise Service Bus (ESB), which facilitates the communication between services. ESB provides all the abilities to communicate, which are essential in calling all the services executed in the entire architecture [4].

Figure 2. SOA reference architecture

Through SOA reference architecture, we can find a complete set of services oriented to users, processes and information integration:

- Interaction services: connection to people in the SOA. They distribute IT functions and data to end users and meet their custom needs.
- Process services: provide orchestrating to interaction between process management and other various services, by accomplishing the business process.
- Information services: facilitate unification, replication and transformation of data. They can be implemented by various means.
- Access services: Achieve existing application assets and data from ESB.
- Partner services: provides documents, protocols and partner management required in the business process.
- Business application services: real-time service for newly-integrated application components in the system.
- Development services: tools are basic components for any architecture.
- IT service management: involves abilities related to the scale and performance.

III BUSINESS PROCESS MANAGEMENT

Business Process Management (BPM) is an IT application that provides operation strategy and solutions which promotes the integration between people, between systems and between people and systems corresponding to the changes of business environment.

Figure 3. Business information system without BPM[5-2]

It consists of several processes such as modeling, implementation, monitoring and management.[12] Figure 3 illustrates a typical IT system in an enterprise nowadays.

The enterprise has already built many application systems, each having their own data, process and user, but isolated from each other.

Enterprise information system with BPM architecture is illustrated as Figure 4. The BPM system, as a work flow center, puts the separated work flows in all the systems together to make an integrated business process, which is standardized and easy to manage.

Figure 4. Flow chart of an implementing enterprise [6-2]

A. BPM workflow

BPM, based on workflow technology, enterprise application integration and XML technologies, manages the enterprise from the perspective of business processes. Also, it supports continuous improvement of business processes, and its core idea is to provide a unified business processes modeling, execution and monitoring environment for enterprise. BPM addresses how organizations can identify, model, develop, deploy, and manage their business processes, including processes that involve IT systems and human interaction [5]. Workflow technology describes the order of a set of tasks performed by various person and/or agents to complete a given procedure within an organization. With the development of management theory and information technology, this kind of static workflow mode has not adapted to rapid changes in demand in business processes. In order to meet the needs of dynamic business processes, BPM has been put forward. BPM not only covers the modeling of business processes, operation, monitoring, analysis, optimization and restructuring, but also plays a supporting role for parallel operation, collaboration and distribution of business processes.

Figure 5. Business process management system

B. The Use of Business Process Management

BPM promises time and cost saving. This environment, a so called process laboratory will provide tools for analysing, designing, implementing, and simulating the Business Process. This process laboratory should make it possible for business analysts to perform all the above activities. These four challenges are discovery, design, development, and deployment of Process. These challenges can be compared to the traditional software development model, which contains the phases of analysis, design, implementation and evaluation.

When a company eventually has gained knowledge of its Process, it is essential that these are represented formally, often visually, to ensure a consistent understanding of the different Process. To be able to automate and thereby optimise the Process within a company, the Process must be designed to run on a Business Process Management System (BPMS). Such a system interprets and executes Process that are fit to be automated and described in one of the formal languages like WS-BPEL. WS-BPEL has a visual notation called Business Process Modelling Notation (BPMN), this notation is used to describe Process [13].

In short, the challenge of the design is to describe the BPs in a standard process language which can be used by business analysts, and not just developers. As in other fields of software development there is a gap between the implementers and the users/customers in their understanding of the system. This is also the case with the development of automated BPs. The business analysts and the developers often use different terminologies, which often leads to automated BPs that do not execute as the original requirements state. Therefore it is recommended that the business analysts use a notation like BPMN and convert this to a process language like WS-BPEL.

Finally, the deployment of a BP can be monitored to ensure that it over time continually conforms to the specification and performs as expected. If this is not the case it is possible to iterate through the four challenges again and determine whether the BP is implemented optimally or it needs to be changed entirely.

C. Composition of web services

The idea of Business Processes based on web services stem from the thought that a Business Process can be composed from one or more web services. There exist two major perspectives of composing web services from one or more service providers as one Business Process, one is called orchestration and the other choreography.

D. Orchestration

The business logic must be distributed into different web services. These services must be organized and composed,

which is commonly referred to as an orchestration [6]. As shown in the fig. 4 an orchestration defines the sequence and conditions in which one Web service invokes other Web services in order to realize some useful function, that is, an orchestration is the pattern of interactions that a Web service agent must follow in order to achieve its goal.

Figure 6. Process of orchestration

Service orchestration should be flexible and adaptable to meet changing Business Process requirements. Separating the business process logic from the service components promotes flexibility. For instance if an organization wants to change the internal details of a service component this should be possible as long as the service's interface remains unchanged, also this should not affect the BPs specified in the organization. By applying this pattern of loose coupling it is possible to swap individual services without introducing system-wide changes in the SOA. The logic behind orchestration is based on rules, conditions, and events.

E. Choreography

At design-time choreography ensures interoperability between a set of peer services from a global perspective, meaning that all participating services are treated equally, in a peer-to-peer fashion. A choreography model describes multiparty collaboration and focuses on message exchange; each Web service involved in a choreography knows exactly when to execute its operations and with whom to interact. [7]

The purpose of using several choreographies simultaneously is to establish a cross-enterprise organized way of collaborating with services. Service choreographies concern the interactions of services with the collaborating users. When a transaction takes place between services it must be defined at the time of execution. Such transactions may consist of multiple separate interactions between the collaborating parties. This means that a single choreography describes the peer to peer interaction in a global, or top down manner. As shown in the fig. 5 composition of web services, the message protocols, interfaces, sequencing, and logic associated with the transaction at hand, are considered to be parts of one choreography [9].

Figure 7. process of choreography

Within choreography it is important to notice that no single enterprise necessarily controls the collaboration logic. The logic can instead be controlled by a number of collaborating enterprises, each having participated in defining the choreography protocol [15]. This provides the potential for creating interoperability patterns expressing common collaborative business processes. When services communicate in a single choreography they each assume a predefined role, which can be specified in the service's interface description. The description of the role defines what kind and in what order messages must be exchanged to perform a given BP. This means that the interaction between the participating services is captured in the formal description. Every single action performed within the choreography is made by a number of message exchanges transmitted on a channel between two services both having assumed an appropriate role in the choreography [10]. Every exchange constitutes a relationship between two services, yielding choreography between them. Choreographies are described using a choreography description language, e.g. WS-CDL.

F. Difference between orchestration and choreography

- Orchestration describes a process flow between services from the perspective of one participant (centralized control), whereas choreography tracks a sequence of messages involving multiple parties (decentralized control, no central server), where no one party truly owns the conversation,[8]
- Orchestration a bottom-up approach to design. Whereas choreography a top-down approach to design
- When multiple processes interact with each other, orchestrations describe the point of view of a single process only. When multiple processes interact with each other, choreographies describe the interplay between processes from a global perspective.
- Orchestration is well understood, Whereas choreography an open research field

Some useful guidelines for understanding which approach is most appropriate for a given scenario are:

Use orchestration:

- When off the shelf products are required (since this is the approach implemented by most current products)
- For composed services

- Where transaction semantics can be handled by compensation alone
- For relatively static process definitions
- Where a graphical process definition is desired

Use choreography:

- Where processes may scale to a very high number of component steps
- Where opacity of process details is desired among process partners (such as B2B)
- Where different process partners may require their own process customizations
- Where processes are highly dynamic or goal-seeking

G. Value of SOA and Business Process Choreography

- Increased automation and deeper process integration, as made possible by the introduction of a choreography component, lead to reduced manual effort and rework both in the wholesaler and the customer processes.
- Process and business rule agility leads to product and process changes being implemented faster and cheaper. Legitimate use of telecommunication resources can be ensured more easily, operational efficiency can be improved, and wastage reduced.
- A variety of interaction styles can be provided to customers while still ensuring consistent application of business rules.
- Reuse of shared application components is simplified.
- Standardization allows replacing unsupported components with commercial-off the- shelf software, which promises to simplify deployment and maintenance of the order management application.

H. Choreography Languages

Choreography languages are WS-CDL, Let's Dance, and BPEL4Chor.[11]

The WS-CDL is the proposed standard for service choreography. WS-CDL choreographies are tightly bound to specific WSDL interfaces; WS-CDL has no multiparty support, no agreed formal foundation, no explicit graphical support, and few or incomplete implementations.

Let's Dance [39] is a language that supports service interaction modeling both from a global and local viewpoint. In a global (or choreography) model, interactions are described from the viewpoint of an ideal observer who oversees all interactions between a set of services. Local models, on the other hand, focus on the perspective of a particular service, capturing only those interactions that directly involve it. Using Let's Dance, a choreography consists of a set of interrelated service interactions which correspond to message exchanges. Communication is performed by an actor playing a role. Interaction is specified using one of three Let's Dance constructs: precedes; the source interaction can only occur after the target interaction has occurred, inhibits; denotes that after the source

interaction has occurred, the target interaction can no longer occur, and finally, weak precedes; denotes that the target interaction can only occur after the source interaction has reached a final status, e.g., completed or skipped.

BPEL4Chor is a proposal for adding an additional layer to BPEL to shift its emphasis from an orchestration language to a complete choreography language. BPEL4Chor is a simple, collection of three artifact types: participant behavior descriptions define the control flow dependencies between activities, in particular between communication activities, at a given participant. A participant topology describes the structural aspects of choreography by specifying participant types, participant references, and message links; this serves as the glue between the participant behaviors descriptions. Finally, participant groundings define the technical configuration details, the choreography becomes Web service specific, concrete links to WSDL definitions, and XSD types are established. BPEL4Chor encourages reuse by only providing a specific Web service mapping in the participant grounding. Furthermore, unknown numbers of participants can be modeled, not possible with WS-CDL.

I. The procedure of designing an architecture frame combining SOA and BPM

The procedure of designing an information system combining SOA and BPM is usually as follows:

Step 1: Define data service and process model

To provide reusable services with a SOA, first define the data, services and process model in the business domain.

Step 2: Define atomic services

Atomic services are those services that can no longer be divided. Designate the size of an atomic service by its reusability.

Step 3: Define composite services

Composite services refer to the Web services that are composed of other services, for it both have WSDL protocol and are both called by SOAP. Composite services can be created directly by programming, or use Web Service Orchestration and WS-BPEL.

Step 4: Define business process

Business process implements complicated business functions of many steps; it usually involves many participators, including internal users, external users and business partners. The run-time semantics of business process is defined by BPEL(WS-BPEL), while the process engine executes the process task and confirms that business plan is implemented according to the WSBPEL script.

Step 5: Study the means by which the service consumer accesses and uses the Web service

All authorized service requestors, including IT systems, terminal user applications and Web services, should be able to access to any services including atomic and composite, and business process published for Web service any time, instead of being limited within eventually compounded business processes.

J. Enterprise Service Bus

The Enterprise Service Bus (ESB) has become the main driver of data delivery in SOA. This means that the ESB makes use of the techniques and technologies provided by the SOA. Furthermore the ESB also handles the transformation of data as the services should not be concerned with such details of other data formats beyond their own. The ESB thereby makes sure that the data transferred to it will be delivered in the correct format at the consumer.[14]

Figure 1: Transformation of messages within an ESB

1. An external partner sends a message (M) in the format (x) as XML to the enterprise. The message in its format is denoted M_x in the following. A business process (not shown) handles the necessary updates in the needed applications of the enterprise (App 1 and App 2).
2. The ESB makes sure that M_x is transformed to the canonical format of the enterprise (M_c). Furthermore the message is made persistent. The message is now represented by M_c .
3. When applications connect to the ESB they must register what format they expect to receive messages in. Thereby the ESB has knowledge of what format to transform a given message into, in this case M_y .
4. The message is passed on to the next application (App 2) which needs to operate on the message. This application understands the canonical format and thereby needs the message to be formatted accordingly.
5. The output from App 2 is formatted in the canonical format, as App 2 internally handles this format.
6. However, before returning a message back to the external partner the ESB must format the message to fulfil the M_x format.

IV Conclusion

BPM is today the preserve of large organisations. BPM, based on workflow technology, enterprise application

integration and XML technologies, manages the enterprise from the perspective of business processes. Also, it supports continuous improvement of business processes, and its core idea is to provide a unified business processes modeling, execution and monitoring environment for enterprise. Much work is ongoing in large companies breaking down silo systems and integrating them using SOA. Combining SOA and BPM can bring complementary advantages. The joining point of SOA and BPM is “service”. SOA provides services of standardized interface by abstraction, and BPM consumes these services in instructing the creation of business process. Orchestration describes a process flow between services from the perspective of one participant (centralized control), whereas choreography tracks a sequence of messages involving multiple parties (decentralized control, no central server), where no one party truly owns the conversation. The collaboration between service abstract layer and business process layer indicates a surprising increase in flexibility and efficiency. This is the greatest motivation of this solution of architecture combining SOA and BPM unified with technology by service.

References

- [1] Eric Newcomer, Greg Lomow: Understanding SOA with Web Services, Addison Wesley, 2005.
- [2]Gopala Krishna Behara: BPM and SOA: A StrategicAlliance,<http://www.bptrends.com/publicationfiles/05%2D06%2DWP%2DBPM%2DSOA%2DBehara%2Epdf>
- [3] An Integration Research on Service-oriented Architecture (SQA) for Logistics Information System Luyang Zhang, Jiaqi Li, Ming Yu 2006 IEEE
- [4] A SOA-BPM-Based Architecture for Intelligent Power Dispatching System Liang Zhang, Shudong Chen, Minhui Ge, Xiaoliang Bi
- [5] A SOA-BPM-Based Architecture for Intelligent Power Dispatching System Liang Zhang, Shudong Chen, Minhui Ge, Xiaoliang Bi 2010 IEEE
- [6] Insights into Web Service Orchestration and Choreography Florian Daniel, Politecnico di Milano, Italy Barbara Pernici, Politecnico di Milano, Italy
- [7] Choreography Design Using WS-BPEL Oliver Kopp Frank Leymann
- [8] Mark Colan, “Service-Oriented Architecture expands the vision of Web services”,<http://www128.ibm.com/developerworks/library/wsoaintro.html>, IBM, U.S., 21 Apr 2004.
- [9] W3C. Web services choreography requirements 1.0., http://www.w3.org/TR/2003/WD-ws-chor-reqs-20030812/#What_is_Web_Services_Choreography
- [10] Nitin Bharti: Dancing with Web services: W3C chair talks choreography,http://searchwebservicestechtarget.com/qna/0,289202,sid26_gci1066118,00.html, 2005
- [11] Choreographing Web Services Adam Barker, Christopher D. Walton, and David Robertson
- [12] **Architecture of Information System Combining SOA and BPM** Yang Liu 2008 International Conference on Information Management, Innovation Management and Industrial Engineering
- [13] Laury Verner: BPM: The Promise and the Challenge, ACM Queue, v.2 n.1, March 2004.
- [14] David A. Chappell: Enterprise Service Bus - Theory in Practice. O'Reilly, 2004.