Enhancing SOA Integration through Enterprise Services

Venky Shankararaman and Lum Eng Kit

School of Information Systems, Singapore Management University
Singapore, Republic of Singapore

Abstract. Service enabling packaged applications can provide greater flexibility and agility to business processes and thus can enhance SOA integration. In this paper we demonstrate using SAP tools a process driven composite application namely, Web Sales Process that leverages "enterprise services" from within the SAP ERP.

1 Introduction

A recent trend among packaged application vendors such as SAP and Oracle is the "service enablement" of their applications through enterprise services. As a result of this "service enablement", a new trend in solution development has emerged namely "Composite Applications". Composition is an emerging approach to delivering enterprise solutions by assembling functionality from prebuilt components. This is analogous to getting a prefabricated house assembled rather than having it custom-built. With the emergence of the composite application paradigm the emphasis is on using model driven and pattern based tools to design and execute the composite application through a graphical interface rather than writing code [1], [2].

In this paper we demonstrate using SAP tools, the use of enterprise services to enable a process driven composite application. In Section 2, we define an enterprise service. Section 3, describes the Web Sales Process. In Section 4 we then present the composite application architecture for this process along with implementation details. Section 5 presents some concluding remarks and future trends in this area.

2 Enterprise Services

An enterprise service is technically a standards based web service that has been purposefully designed for reusability and relates to a business object within the context of a business process. For example, "Create Purchase Order" is an enterprise service that relates to the business object "Purchase Order" in the context of the "Procurement Process" within an ERP. Therefore, an enterprise service typically contains the "best practice" business logic for executing standard business functions. This was traditionally available only through proprietary APIs such as BAPI (Business Application Programming Interface). A BAPI is a proprietary API method of SAP business object. In contrast to a BAPI, an enterprise service uses a web

Shankararaman V. and Eng Kit L..
Enhancing SOA Integration through Enterprise Services.
DOI: 10.5220/0003554600370040
In *Proceedings of the International Joint Workshop on Information Value Management, Future Trends of Model-Driven Development, Recent Trends in SOA Based Information Systems and Modelling and Simulation, Verification and Validation* (RTSOABIS-2011), pages 37-40
ISBN: 978-989-8425-60-7

service based standard interface to access business objects such as “Purchase Order”, “Sales Order”, etc. In the current project, we therefore leverage the enterprise services.

3 Web Sales Process

We describe the Web Sales Process for an automotive spare parts manufacturer. Following is a description of this business process. Fig 1 shows the Web Sales Process, also referred to as the process model using the BPMN standard notations [3].

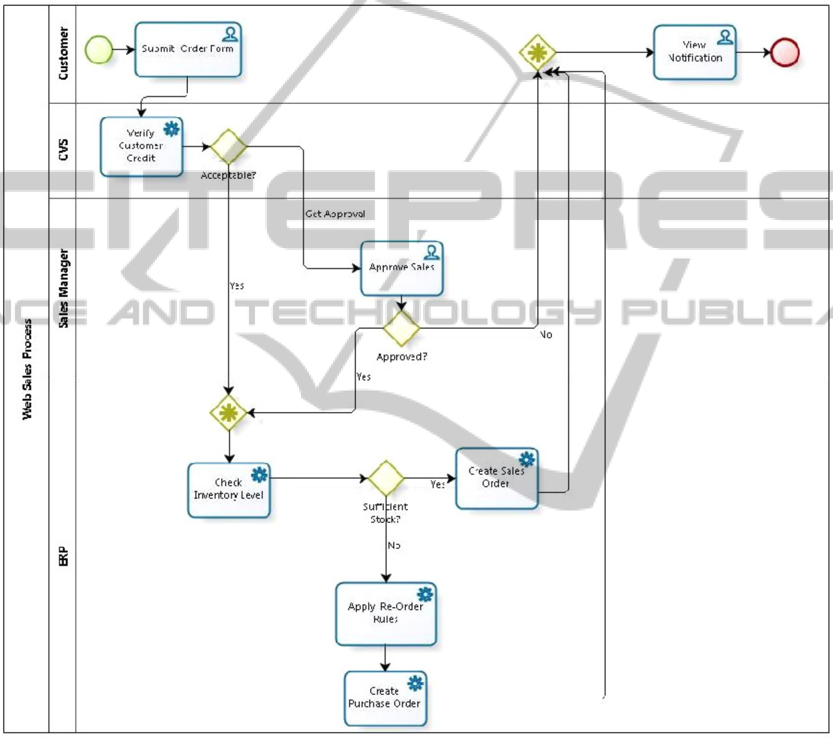

Fig. 1. Web Sales Process.

The customer submits the order by filling an online form. The next activity in the process is to verify the customer credit status.

This is achieved through the Credit Verification System (CVS). The CVS is a Java application that exposes a web service. This web service takes in the Customer ID as input and returns whether additional approval is required. The CVS uses some business logic such as customer loyalty, customer account balance, etc. to determine the credit worthiness.

If approval is required the process waits for the Sales Manager to approve the sales. If no approval is required, availability of the product is checked by invoking the ‘Check Inventory” enterprise service in the ERP.

If sufficient quantity is available, the sales order is created by invoking the “Create Sales Order” enterprise service in the ERP. If sufficient quantity is not available, appropriate business rules are triggered to define re-order quantity. A purchase order is then created by invoking the enterprise service "Create Purchase Order". Depending on the process sequence appropriate notifications are sent to the customer.

4 Web Sales Composite Application Layered Architecture

The Web Sales Process is implemented as a composite application using SAP Netweaver Composition Environment 7.2. The composite application is divided into five layers namely Portal, Process, User Interface (UI), Business Logic and Back End Connectivity (see Fig 2). Table 1 summarizes the different layers along with the SAP tools that were used for the implementation of each layer. For a detailed overview of the different layers one can refer to [4].

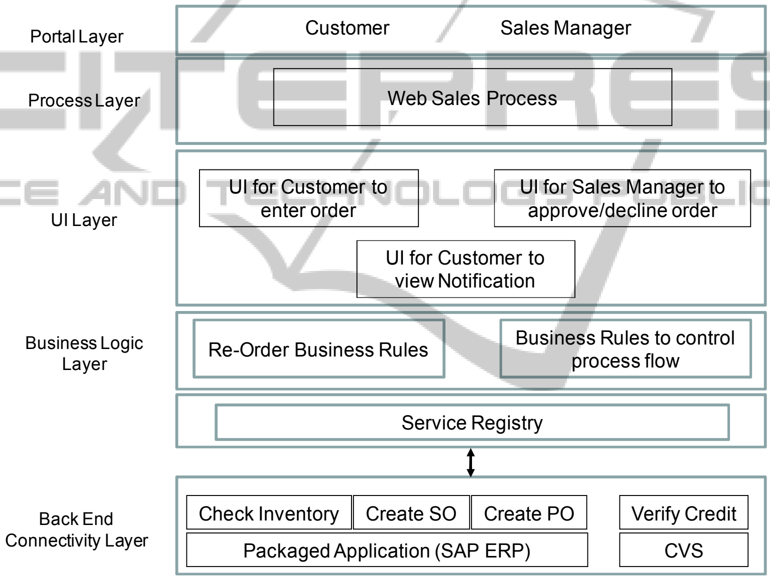
Portal Layer
Customer
Sales Manager
Process Layer
Web Sales Process
UI Layer
UI for Customer to enter order
UI for Sales Manager to approve/decline order
UI for Customer to view Notification
Business Logic Layer
Re-Order Business Rules
Business Rules to control process flow
Service Registry
Back End Connectivity Layer
Check Inventory
Create SO
Create PO
Verify Credit
Packaged Application (SAP ERP)
CVS

Fig. 2. Web Sales Composite Application: Layers.

5 Concluding Remarks

Composite applications enable the creation, delivery and execution of enterprise applications by leveraging existing functionality through service calls that are loosely coupled to backend systems such as ERP through service-oriented interfaces. In a process-based composite application, an individual step within a business process uses the service call to invoke a specific functionality within a backend system. In future, enterprise applications will be more modularized with functionality being exposed as enterprise services. This will lead to more flexible and agile business processes that will extend beyond one organization and thus paving the way for business network transformation [5].

Table 1. Layers and Tools for Web Sales Composite Application.

Layer	Description	SAP Tools
Portal	This layer provides an interface for the roles to interact with the process activities. This is achieved through the portal interface that can provide a single point of access for managing tasks for a role.	Universal Work List-SAP Portal
Process	In this layer, we define the following: Process activities and the sequence in which they are executed, Roles that perform the activities and Business Documents (Data) that is used or generated by the process	Process Composer
User Interface	In this layer, we define the user interface for each activity that has to be performed as part of the business process. The user interface models developed are compiled into General Modeling Language (GML) code and then deployed. At runtime, the GML code of the deployed models is interpreted by a runtime engine.	Visual Composer
Business Logic	This layer includes business logic defined through business rules. In addition, this layer also supports the definition of the rules to manage the flow of the process.	Rules Composer
Service Registry	This layer provides information about the enterprise service which includes reference to the services' relevant WSDL metadata and the location of the callable service endpoint.	Service Registry
Back End Connectivity	This layer provides the ability to make the back end systems transparent to the composite application. Web Services are the recommended technology for connecting composite applications to back-end systems.	Enterprise Services

References

1. M. Huvar, T. Falter, T.Fielder and A.Zubev. Developing Applications with Enterprise SOA. SAP Press. 2009, pp.33-49.
2. J.Rauscher and V.Stiehl. The Developer's Guide to SAP Netweaver Composition Environment.SAP Press.2008.
3. Derek Miers and Stephen A.White. BPMN Modeling and Reference Guide. Future Strategies Inc., Lighthouse Pt, FL. 2008.
4. Eng Kit Lum and Venky Shankararaman.Create a Process Driven Composite Application with CE7.2-Introduction. SAP Community Network. 2010, http://www.sdn.sap.com.
5. Business Network Transformation: Strategies to Reconfigure Your Business Relationships for Competitive Advantage edited by J.Word, Wiley, 2009.