

A Working Heterogeneous Order To Cash System using Oracle's Fusion Middleware

**Kirit Goyal, Manoj Agarwala, Amalendu Samanta
Genpact**

This white paper discusses in detail the design and deployment of a Order to Cash system spanning across multiple systems but threaded together by components to the Fusion Middleware. The system is used in a relatively high volume transaction scenario in a spare parts distribution organization.

Introduction

In an organization where multiple levels of Information Technology penetration exists, it is a difficult choice on the part of the IT manager when demands are made upon him to integrate these systems in real time and improve performance of the system. Many options are available for integration including simple FTP, integrating products like Web Methods and now certain products in Oracle's latest application server release.

This paper presents the description of a working Order to Cash system that is still in the development and testing stage. This system uses the Business Process Execution language (BPEL) as the middleware that provides connectivity to the web based store front, collects the order data, filters it and dispatches it to an Oracle ERP system for further processing. The invoice data is then sent to a collections system which then sends data back to the ERP for application against receipts. The immediate benefits were enhanced real time visibility of orders and operational efficiency.

The Case

X Ltd is a selling and distribution organization and sells parts and assemblies that are used by customers in running and maintaining large infrastructure equipment. X Ltd has multiple modes of capturing orders and quotes. Orders and quotes flow in from a web based store front through which the customer can login and place his request. Other modes of order entry are through EDI, legacy systems and through manual entry. For the Orders and Quotes captured through the storefront, the volume for new orders and updates is an average 1200 transactions per day.

X ltd. has a robust Oracle ERP backbone and all shipping and fulfillment is carried out through Oracle. However, the Web Store front is also a robust application and has been in use for several years with lot of investment and logic built into it.

The requirement for X ltd. Was to be able to flow orders seamlessly and in realtime from the store front to Oracle Applications. A sales order should then be created in Oracle and the sales order number should then be available on the storefront for the customer to reference to. This should happen before the customer checks out from the application. Similar is the requirement in the case of any updates to existing sales orders.

There were also two special requirements for all transactions before the shopping cart was submitted for checkout. They were

- Parts are validated using the Oracle Item Master
- Price is always calculated by Oracle

The first requirement was to minimize the mismatch due to incomplete synchronization between the two item masters and the second was to get the best price for the customer without going into multiple iterations.

The Solution

When X Ltd approached us for a solution, multiple options were evaluated that included

1. A EAI based approach using Web Methods as the backbone
2. A batch processing mode using simple FTP
3. A integrated approach using BPEL and it's components available in the release 2 of Oracle's 10AS

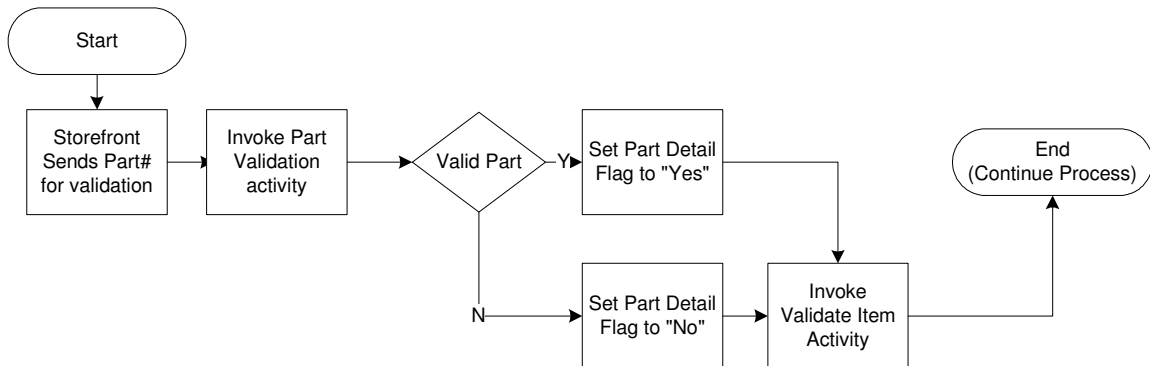
The second option was ruled out as it would not have given a real time solution to the problem. The decision was taken in favor of BPEL over Web Methods primarily because an existing Oracle ERP back bone.

The Figure 3. represents the process flow of the multiple system Order To Cash System modeled using the BPEL designer.

Item Validation Service

The first activity of the process is invoked when the customer enters a part number in the storefront. Because of the nature of the business, it is critical to have the correct part number. Hence there is a need to correctly identify the part number before the order is created. An application on the storefront calls a function that validates the item in the ERP. If a match exists then a valid value is returned. Based on some rules if a number of near matches are returned, the values are presented to the user. He then can select the correct item number. If the item does not exist a value of "UNDEFINED" is returned and the user is prompted to enter extra information.

Fig 1. represents the validate item process.



Call Price Service

The Storefront calls the pricing service to verify the line item prices for valid parts. The Storefront passes the required inputs for each line item. The pricing service returns the price response message to the Storefront. In order to get the price from the ERP, the service calls the QP_PReq_Pub is called by the service. Fig 2. represents the pricing service activity.

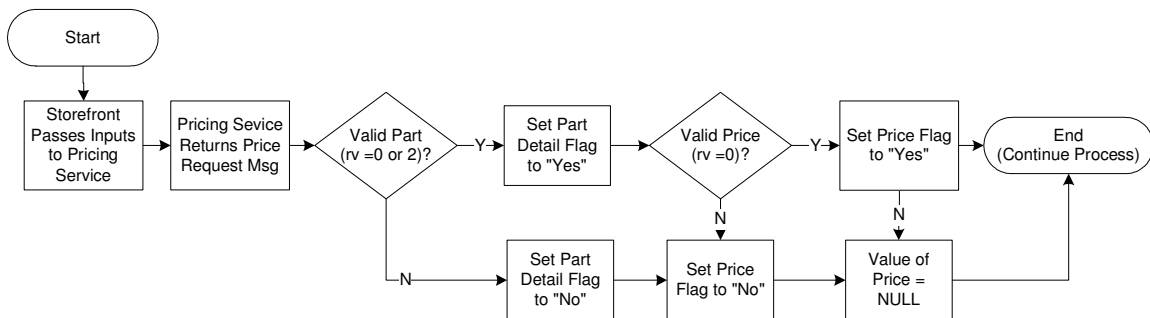


Fig 2. The Pricing Service Process

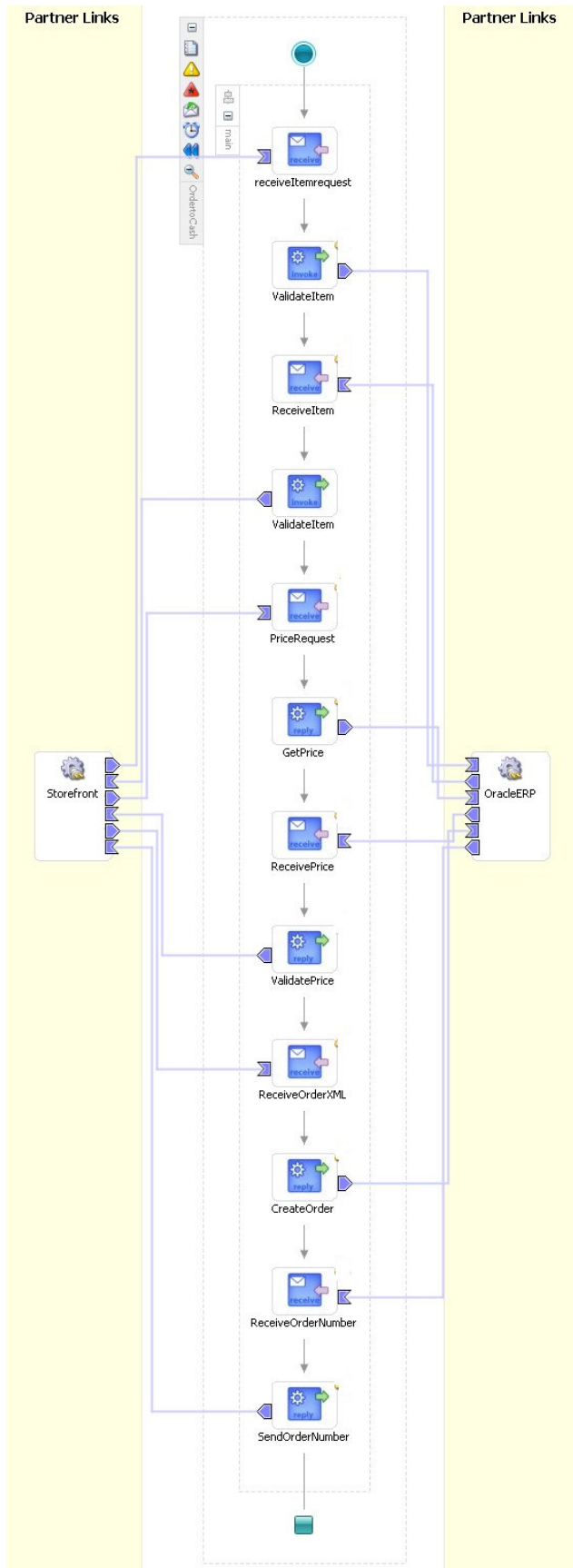


Fig. 3 BPEL Process Flow Diagram

Create Sales Order Service

Once the shopping cart of the user is complete and he wants to submit the quote or the order, he will perform the action in the storefront application. The storefront application will generate an XML file and will transfer the same to the Create Order Service.

The XML file contains all the necessary order header and lines data. The sample of the XML file is in figure 4.

```
<?xml version="1.0" encoding="UTF-8" ?>
= <OrderRequest>
  = <OrderRequestHeader TargetERP="" requestedDeliveryDate="2006-
    01-10" ShipWhenAvailFlag=""
    quoteReferenceNum="0430036/HUK-Kw" probability="Stock"
    customerOrderID="" MixedOrderType=""
    oracleCustomerID="38253" customerID="86-224-8598"
    orderID="231222" orderType="Quote" autoloadingFlag="Y"
    orderDate="2005-11-10" quoteOrderFlag="Q"
    methodOfReceipt="Storefront" priority="Standard Priority"
    type="new" BillOrderType="" Total="" Tax="">
  = <Buyer company="X - Europe">
    = <Address AddressLine1="" AddressLine2="" City="" State=""
      Country="" Zip="" ISOCountryCode="">
  = <Item TargetERP="Oracle ERP" customerLineNum="1"
    condition="NEW" quantity="1" externalCustomerLineID=""
    requestedDeliveryDate="2006-01-10" priority=""
    calculatePriceFlag="Y" lineType="Ship Line Type">
    <PartNum>314A5051P001</PartNum>
    <ManufacturerName>X Services US</ManufacturerName>
    <PartDesc />
    <ProductLineName />
```

Fig 4. Sample XML file transferring data to Create Order Service

The Create Order Service takes the data from the XML file and then passes them as input to the CREATE_ORDER_API. The create order API is then submitted and the sales order is created. If the order creation is successful, the order number is sent back to the Storefront in real time.

Conclusion & Going Forward

The above design has been deployed in a development phase and has been tested for certain types of quotes and orders. While the transfer of data between the systems is seamless, a lot depends upon the quality of data that is passed on by the XML to the create order API. Another issue is the licensing cost of using the 10g Application server. A Jboss environment is cheaper though it brings in another system into the IT foot print.

ERROR: invalidrestore
OFFENDING COMMAND: restore

STACK:

```
-savelevel-  
-savelevel-  
/ANNpdfmark  
/Link  
/Subtype  
-dictionary-  
/Action  
[0 0 0 ]  
/Color  
[0 0 0 ]  
/Border  
[90.0 591.85 96.0 606.85 ]  
/Rect  
-mark-  
/ANNpdfmark  
/Link  
/Subtype  
-dictionary-  
/Action  
[0 0 0 ]  
/Color  
[0 0 0 ]  
/Border  
[102.0 578.8 108.0 593.8 ]  
/Rect  
-mark-  
/ANNpdfmark  
/Link  
/Subtype  
-dictionary-  
/Action  
[0 0 0 ]  
/Color  
[0 0 0 ]  
/Border  
[114.0 468.55 120.0 483.55 ]  
/Rect  
-mark-  
/ANNpdfmark  
/Link  
/Subtype  
-dictionary-  
/Action  
[0 0 0 ]  
/Color  
[0 0 0 ]  
/Border  
[126.0 455.5 132.0 470.5 ]  
/Rect  
-mark-  
/ANNpdfmark  
/Link  
/Subtype  
-dictionary-  
/Action  
[0 0 0 ]  
/Color  
[0 0 0 ]  
/Border  
[114.0 430.3 120.0 445.3 ]  
/Rect  
-mark-
```