

# Implementing Oracle's EBS in a Steel Service Center: An Overview

Wolfgang Moritz, Sr. Business Analyst

*Farwest Steel, Eugene, Oregon*

## Abstract

The Metals Industry is a relative newcomer to using Oracle Applications. Balancing industry conventions, user needs and customer requirements within the existing EBS framework poses significant challenges. Customers require stock goods to be quoted using dollars per hundred pounds or hundred linear feet but billed per each; parts consisting of material bundled with value added services are quoted per each; vendors quote in dollars per ton; inventory is tracked per each, but metrics are measured in tons.

In the fall of 2001 Farwest Steel, a service center located in Eugene, Oregon, undertook this challenge as one of the early adopters of Oracle Applications in the Metals Industry. This paper recounts some of the issues we faced, and discusses the resulting decisions and their outcome. As of December, 2007, we have been live 2 years and are now starting to look at leveraging our E-Business Suite foundation with new Oracle Applications.

## Farwest Steel: An Overview

Farwest Steel is a Steel Service Center, headquartered in Eugene, Oregon. Established in 1956, Farwest Steel has steel, rebar, as well as general fabrication branches in Washington, California, Oregon, Idaho and Utah. These 15 branches, plus our main offices, employ about 750 people.

For nearly 30 years, Farwest relied on a legacy IBM mainframe running a custom "green screen" order fulfillment application. Although this system was well outdated by the time we actually went live on Oracle's E-Business Suite, historically Farwest has embraced new technologies in the warehouse and in the office. By 2000 it was clear that we needed to replace the legacy system and upgrade to a system that could support future growth, more detailed transaction tracking, and increased visibility throughout the business.

After reviewing various applications, including industry specific applications, we decided to implement a full Oracle E-Business Suite – despite the fact that no one else had at that point successfully implemented Oracle Applications in the Metals Industry (to our knowledge). The basis for selecting Oracle over other industry specific applications revolved around a few key points:

- Growth: Anticipating future growth as Farwest acquires new companies and opens new branches; Oracle was a well established enterprise grade application.
- Extensibility: Although we are primarily a Steel Service Center oriented organization, future growth might take us in different directions, and a steel industry or fabrication specific application would only limit us.
- Long term support: As a relatively conservative company, we believe Oracle would be around to provide longer term support.

Each division (steel service center, rebar and fabrication) has unique business requirements, summarized in the table below.

	Steel Service Center	Rebar	Fabrication
<b>Primary Business</b>	Stock (“Structural”) steel distribution, plus some value added services, mainly to OEM’s, other fabricators	Ready to tie rebar for private construction projects, government contracts	Complex assemblies to nearly complete products for retailers, OEM’s
<b>Quoting UOM</b>	Varied: CWT (Dollars/Hundred Pounds) CFT (Dollars/Hundred Feet) CSFT (Dollars/Hundred Square Feet) Each	Lump sum bid on Jobs. Otherwise per pound.	Each (completed part or assembly)
<b>Shipping/Billing UOM</b>	Each	Pounds (to fulfill contract)	Each (completed part or assembly)
<b>Primary Gap</b>	Customer Parts: Bundling material and services into a single billable, shippable item; for hundreds of custom items each day.	Defining customer lump sum Jobs, and tracking Job fulfillment.	Complex assemblies, with many levels of sub-assemblies.
<b>Secondary Gap</b>	Industry conventions for quoting: CWT, CFT and CSFT; but billing per each.		How to define items where the ingredients change with each order?

Among other business requirements we needed to consider during the implementation was costing. Farwest uses Actual Cost (Period Moving Average Cost) instead of Standard Cost, and has made great efforts to track actual costs at a very granular level.

Given these various issues and considerations, the primary application suite selected for implementation consisted of:

- Oracle Process Manufacturing (OPM)
- OPM Financials
- Inventory
- Purchasing
- GL, AR, AP
- Order Management
- Advanced Pricing
- Shipping Delivery Based
- Discoverer
- Off site hosting (presently with Data Intensity)

When the implementation project was approved, Oracle 11i.5.6 was the current release, and Gemms had recently been incorporated into Oracle as OPM. Over the four year implementation many go-live dates were committed to, and were then subsequently pushed back.

Because our initial implementation design context was based on 5.6 functionality, we were faced with many challenges resulting in significant custom development. As our development driven delays mounted, Oracle continued to release new versions. Upgrading to 5.8 without even being actually live added an additional delay; but by the time we were ready to go live on 5.8 the then current release was 5.10, and we resisted an additional delay for yet another upgrade.

It was not an easy implementation; however in the end we successfully went live on 11i.5.8 in December, 2005. Eighteen months later we went live again after finally upgrading to 11i.5.10 CU2.

## Early Decisions: Inventory

Deciding how to define inventory items can be a challenge for any implementation. Other business decisions ultimately revolve around how inventory is setup. OPM nicely handles dual units of measure and actual cost transfer, which is part of the reason we chose this application. Most stock metals goods are defined by their shape, alloy and primary dimensional attribute, but beyond that are fairly interchangeable. For instance, a plate may be 1" thick and consist of a particular alloy, or an angle may be specified as being 4" long per leg and ½" thick across the web; but in both of these cases one or more attributes is needed to define a stocking item: Length, and possibly Width.

Our initial approach was to create items defining the shape, alloy and primary dimension (thickness), and then use Lots to capture the other attributes. However, Oracle OPM did not support Lot Costing at that time which meant that for most unique products the true actual cost would be cloudy. After a few other abandoned concepts, we settled on a brute force approach to handle costing: Define a unique item for each 'stock' size, which encompasses all key attributes. In our legacy system we had smart items, and we continued this approach to setting up our Oracle inventory. Examples of some of our Oracle items:

Item	Primary UOM	Secondary UOM	Description
<b>PL5141/2-48-96</b>	EA	LBS	Plate A514 ½" 48"x96"
<b>PL1/2-REM</b>	LBS	LBS	Plate A36 ½" 1 LBS
<b>A61/2-20</b>	EA	LBS	Angle 6" x 6" x ½", 20'
<b>T84180-40</b>	EA	LBS	Tubing 8"x4"x0.180", 40'
<b>8615</b>	LBS	LBS	#8 Rebar Grade 615

We created an item for nearly every common industry combination of product, alloy, thickness and size, and encoded those attributes into the item number (the resulting item numbers were very close to those from our legacy system, which made the transition somewhat easier).

Given the range of products the typical service center might conceivably carry, this expanded item list amounted to 175,000 unique items in our OPM Item Master. Furthermore, we needed to account for skeletons that had been cut into customer parts or smaller non-standard sizes. For each base combination of product, alloy and thickness we created a '-REM' variant, representing a per pound inventory. A two inch thick 96" x 240" A36 master plate (PL2-96-240) can be cut into three 48" x 96" smaller stock plates, plus 5227.32 LBS of PL2-REM for use in customer parts.

By defining items at such an atomic level we are able to capture material actual cost per sale with less variation in an open period than if the items were defined more broadly.

## Quoting

Initially the assumption was that sales would enter all orders through the Order Management sales form. However, the standard sales form does not handle pricing in one unit of measure (CWT, CFT, or CSFT) while actually transacting the primary unit (EA). To resolve this, our consultants customized the Sales Order form so that the price entered would be converted from the primary quoting UOM for that product to an EA measure. This way an order line for tubing would automatically convert the entered price from CFT into an EA price; and an order line for a plate would be converted from a CWT value into an EA price. This way the transaction UOM for inventory was always measured as Each.

## Customer Parts

Although the sales order form customization resolved the pricing UOM issue for order entry, we were no closer to understanding how to handle customer parts. In a Service Center, customers often supply drawings to have material processed with various services. In many cases these customer parts are one-offs never to be ordered again; in some cases the parts are frequently reordered but the required services may change based on quantity and production requirements. Plus, we wanted to bundle the ingredients (material and services) into a single shippable product. Oracle's Configurator was not an option with our OPM implementation.

Furthermore, we also needed to be able to track these customer parts and maintain a separate part cost and order history for each customer. Given these requirements, creating unique items for each customer part appeared to be the only solution. Yet at a single branch we might sell and produce 100 – 500 different customer parts per day. Not only did we need to define many new items every day, but also create a Formula in OPM to allow us to create a batch for each of these items. It would be physically impossible for an experienced user, let alone a room full of users, to define this many objects each day.

Lastly, there was no system in Oracle to aid in actually quoting the sell price of these parts. Customer parts are normally quoted as a bundled Each price consisting of material and service sell amounts rolled into the total per part price. We did not want to itemize the constituent ingredient prices on the invoice or packing slip.

## Custom Quoting and Order Entry Solution: Keebler

With these various requirements in mind for stock and parts sales, our developer created a Quoting and Order Entry application that interfaces with Order Management. This J2EE web application had the development name of Keebler, which stuck after go-live. (Imagine the room of the aforementioned users creating items, formulas and recipes as elves doing the heavy, behind-the-scenes work in Oracle.)

This custom application allows users to easily quote stock items using the industry standard UOMs: CWT, CFT, and CSQF. The user interface has been optimized for the Steel Industry, and allows users to quickly enter, modify, and track quotes. Keebler maintains a synchronized customer and item master, so to the end sales user Keebler has everything they need to quickly conduct business, including beautifully formatted quotes that can be emailed or faxed to the customer.

When a customer confirms a purchase order, the quote is converted into an Order Management Sales Order by pressing an "Oracleize" button. Oracle still manages the key processes and transactions: inventory management, order fulfillment, manufacturing, financials etc. However the business area that is highly steel industry centric – quoting and order entry – is optimized for the user.

Keebler's other strength is in allowing processing sales to easily handle customer parts. With a customer drawing in hand, a user can specify that this quote line is a part and enter the primary material plus any number of services

(inside or outside). To help derive an appropriate sell price for the part, the user can also define key attributes such as the number of linear inches to cut, how many pierces, the quality requirements, the vendor if the service is for outside processing etc. Keebler has built in algorithms to convert these service attributes into an estimate of processing time and cost. That, plus material cost, allows Keebler to generate a suggested sell price for each customer part.

When the quote for customer parts is Oraclized, Keebler first creates a new item in the item master, using the customer number plus the supplied drawing name as the item name. Then Keebler creates an OPM Formula and Recipe for this item (using the estimated service times and materials from the quote as the plan quantities in the Formula), and lastly a Sales Order for the newly created item. If the part requires outside processing, a Purchase Order is also created.

With all of the Oracle side documents created, a scheduled Concurrent Request then selects newly created Order Management sales lines not on credit hold and creates a pending batch for that sales order using the appropriate formula.

Keebler only needs to create the item/formula/recipe the first time the part is ordered. To allow for a more flexible and reusable Formula for subsequent orders, our convention is to define all Formulas as a single unit output for the product (and only one product per formula), and all ingredients are in per unit measures. Although this works easily to scale services measured in minutes per unit, the material ingredient is changed from a stock dimension measured in EA, to a REM variant of the same item measured in pounds. This way future orders can be easily scaled.

If the part is ordered by the customer again, the user can “shop” for the customer part in Keebler’s parts library using a form similar to that of online retailers. Customer parts can be added to a shopping cart, and in turn added to a new quote. On these subsequent orders, Keebler only creates the Order Management sales order, letting the Concurrent Program create a new batch for the ordered quantity using the original formula.

By defining unique items for customer parts, we are able to clearly track a part through production, inventory and shipment. Furthermore, we can robustly report customer part order history and costs, among other transaction details. Also, parts are invoiced as a single sales line, showing the each and total price for the customer.

## Rebar Jobs

The majority of our rebar sales are for specific and significant projects, as opposed to random sales. Most of these jobs require us to supply a lump sum quote with an itemized list of materials, broken down by each construction stage, called the ‘Bid Item’. Then, if the contract is awarded, the job is fulfilled over months (or years in some cases), and both the customer and our rebar staff carefully monitor the actual sales against the original estimate.

For reasons not fully understood anymore, we decided not to use an Oracle application such as Projects. Mostly likely this was due to limitations in the 11i.5.6 instance on which our implementation was initially based. Instead, we developed a custom solution using mostly seeded forms and behaviors.

To allow us to maintain separate billing and aging for a customer’s different jobs, plus potential lien rights on each job, we chose to create a new customer account for each awarded job under that customer’s party entity. Some customer parties may have dozens of customer accounts representing the different jobs.

Because the next most granular entity after customer account is the customer site, we elected to define each construction stage (Bid Item) as a separate Ship To site, all pointing to a common Bill To.

With the customer account defined for a specific job, rebar sales could then enter sales orders, one for each Bid Item (Ship To) within the project, using a specific Order Type named, “Quote.” (Quotes did not exist in 5.6.) These

Quote Order Types were not to be booked and transacted, but were records for the original estimate for each job. To prevent booking a Quote Order Type, a custom workflow was created. Later, this custom workflow was replaced with a simple Forms Personalization.

After entering a set of Job Quotes in Order Management, actual sales orders could then be placed using the specific customer job account and Bid Item. To maintain per unit pricing consistency, the quotes and orders both reference a specific Price Agreement created for each job.

As the job progresses, a set of reports are used to compare the originally quoted material quantities and total sell with the actual quantities and sell. This table helps to illustrate the concept:

Party: Contractor Entity						
	Customer (Job) 1234			Customer (Job) 5679		
	Quote	Orders	Remaining Q-O=R	Quote	Orders	Remaining Q-O=R
Bid Item (Ship To) 1	100	75	25	500	550	-50
Bid Item (Ship To) 2	200	210	-10	500	450	50
Bid Item (Ship To) 3	300	290	10	100	100	0
Totals	600	575	25	200	200	0

Despite being a non-standard adaptation of the trading community architecture, the system has worked well so far.

## Fabrication Issues

Although Farwest Steel has always promoted fabrication services to customers, at the initial Oracle go-live our material processing was mostly limited to first stage processing using a single type of material with a few services. However, soon after go-live we acquired several full fabrication centers and are now in the process of implementing Oracle Applications in a true fabrication job shop.

After completing the initial gap analysis we concluded that our basic approach to processing would be continued (creating new items for each unique part and sub-assembly, creating batches for each sales order etc.). However, unlike in our service centers, these fabrication parts may consist of complex assemblies, with multiple levels of assemblies and subassemblies with many materials and services being used. Aside from the significantly longer lists of ingredients used in fabrication batches (we estimate up to several hundred items in a typical batch), our current system needed only minor modifications.

## Application Express + K2: The Evolution Continues

Keebler has served us well over the last couple of years. However, we also wanted to open Keebler development to our Business Analysts and consolidate the Keebler database into our EBS database to avoid data replication issues.

We recently discovered Oracle's Application Express, which allows one to create custom applications over Oracle database objects through a friendly UI. The Apex application (and custom user applications) is compiled entirely in the EBS database, and a web server (we're using a Discoverer server) delivers the applications to end users. We quickly recognized the value of running Apex in concert with our EBS, and we are currently migrating Keebler to Apex. This allows multiple Business Analysts to contribute to K2 development (second generation Keebler), and resolves data replication issues. K2 database objects live in a separate custom schema, and all transactions are completed through Oracle public APIs.

As part of the K2 migration, we reviewed our business requirements for the existing business units and the new fabrication centers. Despite the fairly short list of gaps to implement our business solution in a fabrication environment, that gap analysis led us to reevaluate how we use formulas in OPM. Keebler creates formulas based on the initial Oraclization of the part. However, in a job-shop production environment driven by many sales orders, Formula driven Batches are not flexible enough.

On subsequent orders for custom parts the ingredients might need to change based on new conditions. In some cases a different alloy or grade of material may be needed. For load balancing or quality reasons a different set of services may be required. Furthermore, using only REM in our Formulas has produced a cloudy source material to part genealogy. Updating the Formula has proven to be too cumbersome given the volume of transactions, and users tend to create a new item/formula/recipe (using the Keebler system) when the customer requirements change.

Although we considered enabling K2 to update existing Formulas through APIs, we realized that the Formula was still not flexible enough. Our goal was that once an item was created as a new part, the part definition could be fluid from order to order, as conditions and requirements change.

To resolve this, we will be creating 'Formula-less' batches in OPM, using the K2 list of part ingredients from the quote as a surrogate Formula. (Of course a Formula is technically still needed, so a generic Formula named **Production Part**, which has only has placeholder items (**Enter Item**) for both the ingredient and product tabs, is used.) A scheduled Concurrent Request then finds new sales lines not on credit hold, and then creates a batch using the generic **Production Part** Formula. This Concurrent Request then simply inserts the actual product and ingredient lines into the new generic batch from the K2 part ingredient details table.

With this new system in place, users will be able to define the part as needed on the fly for each sales line.

## Other Notable Customizations

Our initial customizations (other than Keebler) were all in the application environment. However, these custom Oracle forms were still somewhat clumsy to use. Because of this, we have focused on custom web based applications outside of the Oracle environment providing a streamlined set of tools to accomplish the specific task at hand. In general, we now believe that any customization that requires a custom form is best developed as an independent web based application.

For the metals industry, OPM's Item Inquiry is an inadequate tool for inspecting inventory, as is Availability in Order Management. Because larger items may be easily substituted (and up-sold) for smaller items of the same grade and type, we needed to see a list of items in increasing size given an initial input. We also wanted to be able to easily inspect the inventory in other branches in case a requisition could fulfill the customer's requirements. Our inventory inspection tool (Zack) accomplishes this as a web application with a link on the front page of our intranet. This tool allows anyone to check inventory, and see a range of items that might fulfill the need.

Also, to manage the steady flow of batches being created by the system, we created a simple web application that displays all pending batches for a given organization. A user may select specific batches (or many batches) by

checking a box. When executed, this application then releases the batches to WIP, and prints a batch ticket (also customized) to a local printer at the same time.

## Going Live

The first few months post go live were very challenging. We had scheduled and completed thorough user training for actual application users. Sales users quickly adapted to Keebler, however operations was harder hit. Perhaps not enough effort was invested in training non-application users. Although warehouse staff might never see an Oracle logon, their job was highly impacted with new documents, workflows, and new expectations for their work.

The transition from our old paper based system to the new system was difficult, specifically in shipping. We did not implement a warehouse management system, so all Oracle inventory transactions left the office as a piece of paper (Pick Slip or Batch Ticket), and eventually returned to the office with operator notes as to what items were actually picked or consumed.

In many steel service centers it is understood by warehouse personnel that sometimes material may be substituted on the fly to fulfill customer needs. For instance, a larger plate can be quickly cut into smaller plates to fulfill a customer order; in the past the warehouse could do this with little oversight. However, this is in conflict with the ERP which depends on the assumption that the entered inventory quantities match actual quantities, and those larger plates are now no longer available despite what the application shows.

Within a few months, operations learned to work with the system and maintain accurate inventory levels through better communication with the office. After two annual inventories, we believe we have a much more accurate count than at any time in the past.

## Future Directions

Now that we are stable on our EBS, we are exploring new applications to leverage our system. The application suite we went live with is providing the foundation for future strategic opportunities.

Applications or areas of focus we are considering:

- Supply Chain Planning and Demand Forecasting
- Warehouse Execution System
- Better inventory management by use of bar codes or RFID devices
- Transportation Management
- CRM
- Business Intelligence, Dashboards, KPIs
- More Apex based custom applications, such as shop floor batch control



## Conclusion

Implementing Oracle's E-Business Suite was ultimately worth effort. If we were to start over again today some of our original decisions would likely change; but in most cases we would make the same choices again. Our EBS is not static, and we expect to refine, and enhance our systems to meet new business challenges.

Wolfgang Moritz  
Sr. Business Analyst  
Farwest Steel