

Oracle Order Management in Logistics Outsourcing Enabling Growth Through Functional Scalability

Andrei Larionov
April 15, 2002

Introduction

Imagine three major software producers as your clients, each with millions of customers. 3000 different SKUs in the warehouse. Thousands of orders to be processed daily. Shipping delays not an option. These are all challenges "Northern Logistics" was facing in the spring of 2001.

In this presentation, you will learn how Oracle Order Management was used to automate NL's logistics outsourcing business, enabling to process 250'000 orders to date, at an average rate of 1700 orders per day.

Our presentation will appeal to the audience who implemented or plan to implement Oracle Order Management in the high transactions volumes environment, with interfaces to external systems.

Business

Northern Logistics (NL) is in essence an extension of its clients' respective businesses. As an integrated service provider, it interfaces with Direct Sales Support, Manufacturing & Fulfillment departments of its clients, in addition to their respective customers. Company's ability to manage these functions effectively and provide clients with timely information about their operation defines its value.

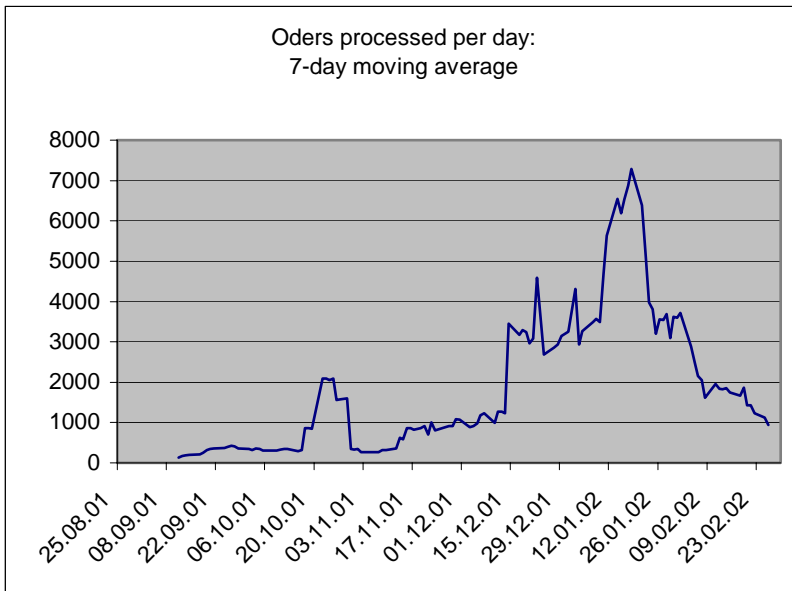
In short, the business works as follows:

Clients fully outsource their warehousing to NL. Daily, they send in batches of orders. NL should pick, pack, and ship these orders based on the shipping priority, carrier type, etc. NL should provide customers with the information about inventory levels and orders processing status.

Requirements

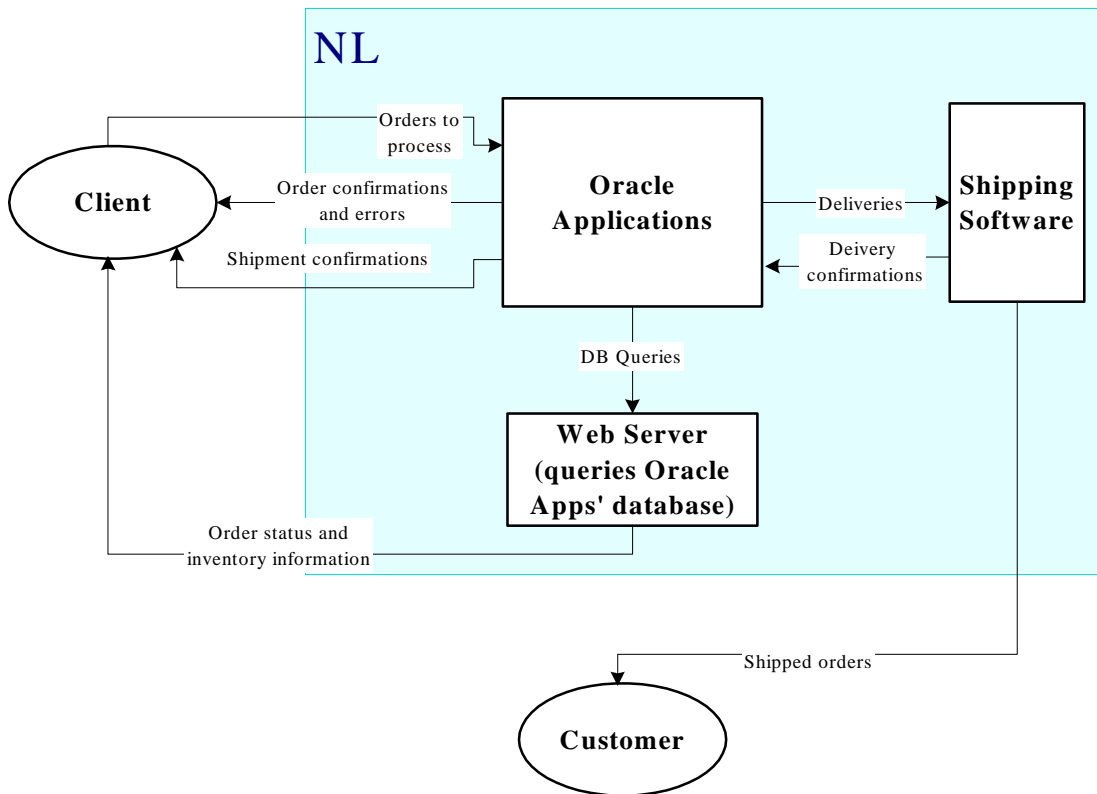
The unique characteristics of this project included high transactions volumes, need for fast order processing, using of batch mode for virtually all order processing steps, and low tolerance for errors and disruptions due to the competition and highly demanding clients.

The graph below visualizes NL's transaction volumes:



Because Oracle Apps had to exchange data with specialized systems, such as shipping and manifesting stations, as well as client's own information systems, heavy use of interfaces was necessary.

Following context diagram shows how Oracle Apps interfaces with external systems



Functional set-up

In order to avoid ambiguities, we will use the following terms in our presentation:

Companies that outsource their operations to Northern Logistics will be called **clients**, while the firms and individuals to whom orders are shipped will be called **customers**.

Following are some highlights about the functionality that we believe were important in our project.

Separating data of different clients:

Our functional set-up had to provide for the full separation of data of different clients. We had to be able to independently process inventory transactions and batches of orders of different clients in terms of inventory, we found that the FlexFields functionality ideally suits our needs here – 1st segment of the SKU number is the Client code, 2nd segment – the SKU number.

Using of pick release batch number:

We discovered that using of the pick release batch number for selecting released transactions when printing the pick slips and interfacing them to external systems is much more convenient than the pick slip number. When you release order in batches, batch is ultimately a unit of work, which should be used to trace the orders released through picking to shipping to ship confirmation. We had to re-program the picking slips to work with the pick release batch numbers rather than with the pick slip numbers

Two-staged picking:

In certain business environments, characterized by processing large volumes of shipments, where SKUs are small in size, it makes sense to produce a bulk picking slip before the actual picking, as it simplifies and speeds up the process. In our case, we first transfer all inventories for a certain batch of orders into a separate location, then, pick from that location.

Using of Ship Sets

Ship sets are very useful tool which, when properly used, adds a significant value to your order processing. You can save processing and shipping costs by avoiding shipping an order in many shipments. Ship set goes beyond “ship in full/ship partial” approach, allowing you to enable / disable partial shipments for order, separate lines, or groups of lines.

Order status inquiry

We were disappointed to discover that Oracle Order Management does not show order statuses in precise and understandable way. In the version 11.5.3 that we used, you could not tell from looking at the order in the “Order organizer” whether it has been released for picking or backordered. Also, you can not branch to deliveries for a given order from the “Orders organizer” screen. Yet we needed to display order statuses on the web page for our clients.

The challenge here was : given that every order line can be in a different stage of processing (backordered, released, shipped, etc..), how to show the order status to the customer in the simple yet informative way.

We decided that following statuses of the order header are enough for our purposes: accepted, in-process, backordered partially, backordered fully, shipped.

Order Processing

developed by Northern Logistics Inc.

Order Lookup
Inventory Management
Order Processing
Accounting
NLPRO

Order Status Report

SKU Number : Customer Order From: NL Order From: Accepted Date From:

To: To:

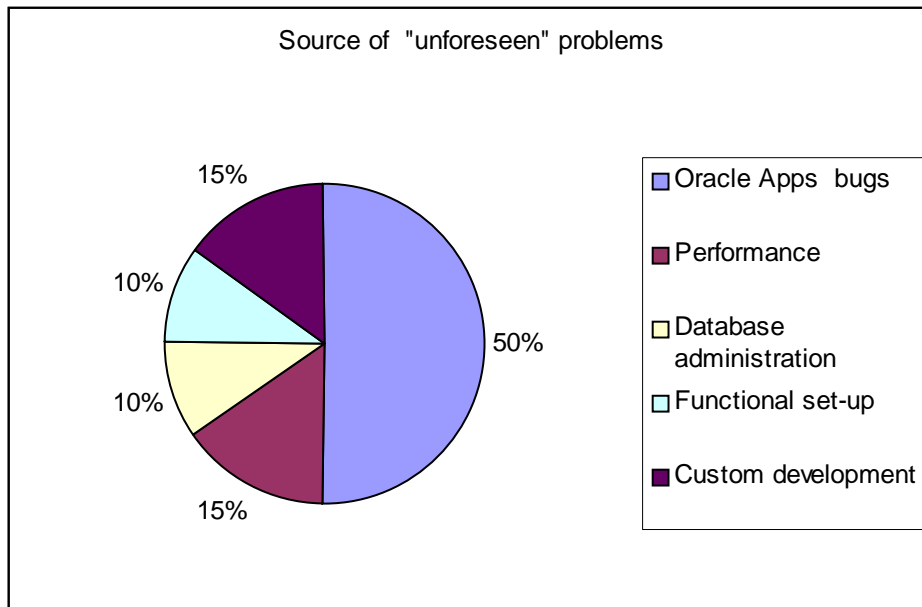
Order Status: Accepted In Process Shipped Fully Backordered Fully Partially Shipped

Client Order #	NL Order #	Accepted Date	Order Status	Last Action Da
1808589	246285	3/8/2002 4:05:13 PM	Shipped	3/8/2002
1808590	246288	3/8/2002 4:05:16 PM	Shipped	3/8/2002
1808593	246289	3/8/2002 4:05:19 PM	Shipped	3/8/2002
1808594	246291	3/8/2002 4:05:22 PM	Shipped	3/8/2002
1808598	246293	3/8/2002 4:05:24 PM	Shipped	3/8/2002
1808599	246294	3/8/2002 4:05:27 PM	Shipped	3/8/2002
1808600	246296	3/8/2002 4:05:28 PM	Shipped	3/8/2002
1808602	246298	3/8/2002 4:05:31 PM	Shipped	3/8/2002
1808603	246422	3/11/2002 10:15:54 AM	In Process	3/11/2002 10:15
1808604	246300	3/8/2002 4:05:32 PM	Shipped	3/8/2002
1808607	246423	3/11/2002 10:15:55 AM	Back Ordered	3/11/2002 10:15
1808608	246424	3/11/2002 10:15:57 AM	Back Ordered	3/11/2002 10:15
1808610	246302	3/8/2002 4:05:34 PM	Shipped	3/8/2002
1808611	246427	3/11/2002 10:15:59 AM	Back Ordered	3/11/2002 10:15
1808612	246431	3/11/2002 10:16:03 AM	In Process	3/11/2002 10:16
1808614	246434	3/11/2002 10:16:06 AM	Back Ordered	3/11/2002 10:16
1808615	246435	3/11/2002 10:16:08 AM	Back Ordered	3/11/2002 10:16
1808616	246438	3/11/2002 10:16:09 AM	Back Ordered	3/11/2002 10:16
1808617	246441	3/11/2002 10:16:11 AM	Back Ordered	3/11/2002 10:16

Because of the problems with order statuses in Apps, we had to develop the customized procedure which defined order status by querying and comparing the quantities in each stage of processing cycle for each order line.

Problems and challenges

It is well known that implementation of any ERP system is full of surprises. The following diagram represents the effort we spent on resolving “unforeseen” issues, i.e., the work that is usually referred to as “contingency” in your project estimate.



Bugs

The number of bugs unpleasantly surprised us, perhaps because we were using earlier versions: 11.5.2, then, 11.5.3. Hopefully, later versions will have fewer bugs.

Early in the implementation, we even had to switch from 11.5.2 to 11.5.3 because of the number of bugs. We believed that re-installing the next version of the system would take less time than applying "family packs", even if you have to re-configure everything. (This opinion was based on experience of applying patches on other projects).

Performance.

We approached performance problems from both hardware and software sides
For hardware, we switched to a more powerful server and RAID 1+0 from Raid 5

For software, our experience shows that you should follow these steps:

First, make sure cost-based optimization is enabled for your database and run the "Gather statistics" – this step can sometimes yield wonderful results. In our case, performance of order import improved 3 times in comparison to the database with no statistics.

Second, apply all latest patches, however painful it is – we achieved acceptable performance of Order Import only after application of the next OM patch set.

Third, fine-tune Oracle Apps – increase the number of child requests for order import, change the parameters of "INV remote procedure manager" etc.

In our case, we were able to achieve more than 50% increase in the performance of Interface Trip Stop process by simply changing the configuration of "INV remote procedure manager". This tuning of parameters requires some expert advice or painstaking research in Metalink.

We would advise everyone with performance problems to begin with tuning your software first. Sometimes, you can achieve fantastic performance improvements by simply tweaking the set-up of apps. This option is also more cost-effective than the hardware upgrade.

Database administration

The database of oracle database comes with “autoextend” disabled for some tablespaces. If a process crashes because it can not extend the database object, it might not roll-back correctly which might cause data corruption.

Right at the go-live, you should make sure to enable “autoextend” for all affected objects. We forgot about this, and had to interrupt using the system for DB admin tasks.

Summary

In the end of the day, Oracle Apps does its job. It is a scalable and robust system, which you can extend with your custom developed reports and interfaces.

Whether you will be able to use in full the opportunities Oracle Apps gives you, depends on the proper decisions about functional set-up at the time of implementation.