

NMHG's use of Agile e6 and the Pro/e CAD Connector

Russell 'Rusty' Mellmer
NACCO Materials Handling Group, Inc.
4000 NE Blue Lake Road
Fairview, OR 97024

Keywords:

Agile e6, xPLM customizations, NMHG Customizations

Introduction

The presentation will focus on NMHG's use of Agile e6 and the customizations that xPLM provided for the CAD Connector as well as NMHG's customizations for Product Development.

Company Background

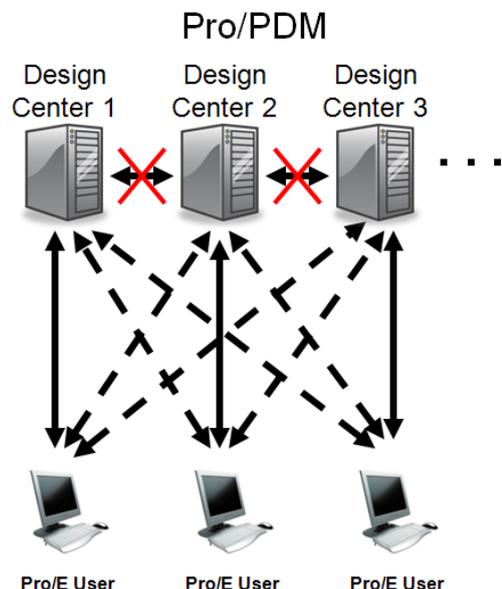
NACCO Industries is an operating holding company with subsidiaries in the lift truck, small appliances, specialty retail and mining industries. In 2011, total revenues were \$3.3 billion. NACCO Materials Handling Group, Inc. (NMHG) is a leading global supplier of Material Handling Equipment marketed globally primarily under the brand names of Hyster & Yale. NMHG manufactures a wide range of lift trucks from small electric hand trucks to large container handling lift trucks.

NMHG Distributed Environment

In 1997 the decision was made to move the CAD maintenance of the vehicles from the 3 main design centers to the plants where the product is built. Before this each design center had separate processes and separate CAD standards. The plant would request a change to the parts and the engineer at the design center would make the necessary changes. Each location had its own database and very little data was shared between the design centers. After the distribution, plants became responsible for the maintenance of the vehicles which required access to the various databases spread throughout the company. Thus access went from the local LAN to the WAN. Distributed engineering has now evolved to a point where engineers can be used as resources for any project at any location.

Management of CAD Data before Agile e6

The issue is that with Pro/PDM we had separate databases spread throughout the company with no connection between the databases. This resulted in duplicate models and issues where users didn't know which model was the correct "master" model. It was also confusing to engineers when connecting to a different database each time to retrieve data.



PLM Selection Process

When Pro/PDM was retired by PTC we decided it was time to start looking for a replacement tool. In the summer of 2004 we started the process and invited seven vendors to participate in a selection process. Four of the seven agreed. The four participants were UGS (TeamCenter), Agile (Agile e6), PTC (Windchill), and Softech (Product Center). The initial need was to replace the Pro/PDM CAD data management tool, but we also looked at the bigger picture of a full blown PLM system. We needed a tool that could handle our distributed environment, (i.e. a central metadata server and local file vaulting) a tool that was easy to configure and customize, had a user friendly interface and one that could handle the Pro/e relationships and large assemblies. Agile e6 was the winner of the benchmark and in early 2006 it was deployed globally.

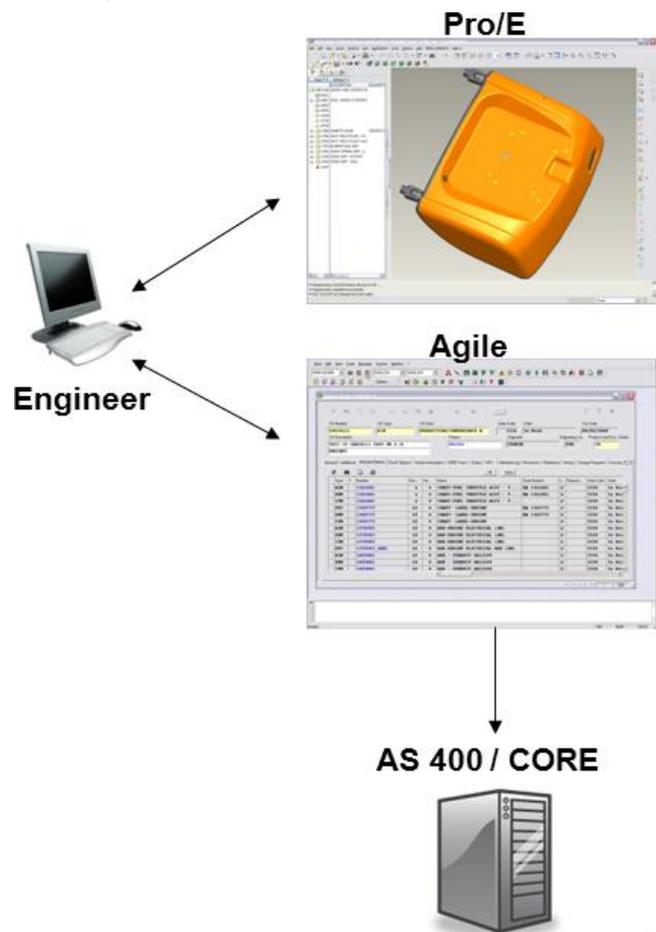
Agile Data Management

With the introduction of Agile, we were able to get rid of all our duplicate models and setup a centralized database with replicated file vaults. NMHG currently has 11 DFM servers globally. A central database and Application server in Fairview, OR. and connections from over 14 locations. Initially Agile was rolled out to handle only the CAD Data. Our in house CORE system was the product data management system for items and BOM information along with historical usage.

We currently have a spiderweb of systems for Product Development, but the goal is to continually move information from our in-house CORE system into Agile e6.

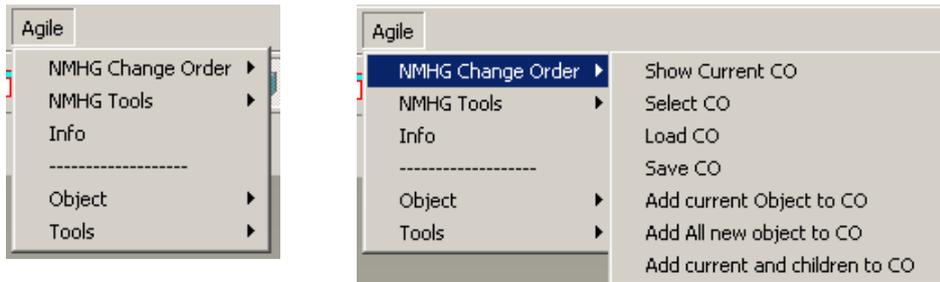
In 2008 we migrated items from CORE into Agile which reduced duplicate entries into different systems. The user now creates the Change Order in Agile as well as the items and BOM information. This is then transferred into CORE which still is the system that passes information to our downstream manufacturing systems. The long term goal is to keep the engineers interfacing with just the Agile system, and we are getting close to achieving that goal.

Recently we moved Units, Master Lists and Planning Lists into Agile. Master Lists and Planning Lists contain the product structure based on sales options. This has improved the engineers experience so they don't have to go between Agile and CORE systems. They can now do all the product development and structuring inside of Agile. We also have the product structured by Engineering functional systems which is related closer to how the engineer designs and the Pro/e assemblies they create. We currently maintain the sales structure (sBOM) as well as the engineering structure (eBOM) in Agile.



xPLM customizations

During the initial implementation, we wanted a system that required users to work on Change Orders in order to control the changes to CAD documents. xPLM created customizations specific to NMHG for working from Pro/e through a Change Order and the ability to add Pro/e documents to a Change Order from inside of Pro/e. The ability to save only the files on the active Change Order and customizations for family table parts was also done by xPLM. All changes in Agile require a CO and the user must have access to the CO in order to save anything into Agile from Pro/e.



Agile e6 with Pro/e

This area will be describing how a typical engineer works with Agile. Below was a major customization that we had xPLM do for NMHG in the way we modify and save Pro/e files into Agile.

1. The engineer creates the CO inside of Agile. All changes are controlled by CO's at NMHG and this is how information is passed between various systems.
2. The engineer works in Pro/e determining what changes are actually required and from there can add Pro/e documents to the Change Order in Agile. This was a customization created by xPLM. This adds the "Released" document to the CO inside of Agile.
3. The user would go into Agile and create a new revision of the document so that it was at an "In Work" status and could be updated from Pro/e.
4. The issue is that the document in Agile (new "in Work") is not linked to the Pro/e part that is active in Pro/e. xPLM created a customization that allows the user to simply save the Pro/e document into Agile. In the background it clears the current link (PLM->User) and then links with the new "In Work" document in Agile and saves the file against this document.

Reference Directory

The purpose of the reference directory is to provide the engineer with the latest changes to the product as "reference" information. The Reference Directory is a list of folders on servers at each location that contain the parts required for each TVA.

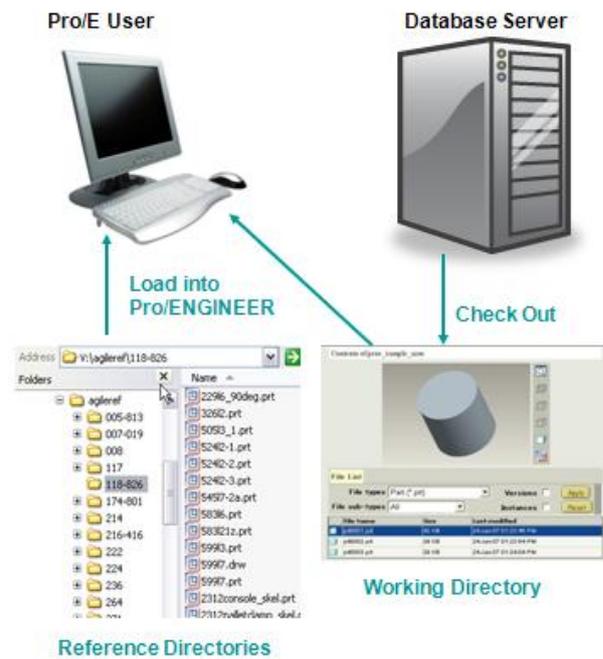
910-187

967-267

976-276

The engineers check out only the parts they are working on. Parts not contained in the working directory are opened from the reference directory via Pro/e search paths. The reference directories minimize the amount of Pro/e data copied out of the database per user to a working directory.

As files are added or removed from a TVA, automated scripts process the various TVAs and either add or remove the files from the folder. The Reference Directory folders are replicated to the appropriate locations that need access to the TVA for design or maintenance. Data from the reference directory is read only, so parts cannot be saved back to the reference directory within a Pro/E session. This was an xPLM customization that does not allow any Pro/e parts in session from the reference directory to be saved into Agile, even if it is on the CO.



Future xPLM customizations

Single Level Structure Save

Many times the user wants to just save the active assembly and update the structure for it. Currently it will do an entire User->PLM for the entire assembly. We'd like to have it just look at the level 1 structure to speed up the save process.

Family Table Improvements

Saving a family table has to go through every instance in order to determine the parameters and also the structure for the instances. We'd like to see improvements where it could look at the family table information to determine the parameters and structure instead of having to open up every instance into RAM in order to get the information into Agile.

NMHG Customizations

Documents and Items - The Pro/e Model contains parameters that update the 3D Model Document as well as the 2D Drawing Document. If the documents are attached to an item, the documents will update the fields on the Items as well.

CO Validations - We have added customizations to Agile for validating a CO and the items and documents that are on it. NMHG has many rules for items such as standard descriptions, stock disposition, service parts, functional divisions, revision rules, etc... All of these validations are checked before a CO can move to the "In Review" phase.'

Affected Objects Tab - User can select multiple items on the CO and click the update icon to update multiple items at once with the same value. Certain fields can only be updated if a document is not attached to the item

Colors - Since the upgrade to Agile e6.1.1 and the move from the Windows Client to the Java Client, we have added various colors to different lists in Agile that indicate warnings or errors, Activity on a part, cancelled items, release levels, etc...

Create BOM from File - We have added the ability to create a BOM from a text file. This is only allowed for specific items where there are no Pro/e documents attached since the BOM creation should come from the Pro/e structure.

Direct Database Connections - We have created connections from Agile to DB2 (our home grown product development system for items and BOM control), a SQL Development Log database for Part Activity and a SQL database for procurement to source new parts.

Issues we had to fix - When we moved items into Agile, we discovered an issue when copying and deleting items from Agile. Many times users will put an item on a CO, copy it to a "new revision" and then decide they do not want to make that change after all. They then delete the "In Work" item and revert back to the "Released" item. The issue we discovered was that if the parent item is copied to "In Work" and then a child of that parent is copied to "In Work", that the old revision of the parent doesn't contain the new "In Work" child item. Thus, if the user deletes the new parent item to revert back to the old item, the current view of the item doesn't contain the newly revised item. It is not on the BOM. In order to fix this issue, whenever a child is copied to a new revision it also adds the "In Work" child to the previous revision of the parent.

Current Enhancements - Effectivity Dates in Agile e6

NMHG is planning to use effectivity dates in Agile to handle a change to our Product Development process. If a specific project is attached to the CO, it will in turn set an Engineering effectivity date that will set the VALID_FROM date for all items released on that CO. The effectivity date is set in the future. The user can then view various revisions and BOM structure changes based on this effectivity date in Agile.

The image to the right shows 4054262 (rev 0) when viewed with an effectivity date of 4/1/2013.

Position	Number	Revis...	Versi...	E	Name
10	4054259	0	0		
10	4054260	2	0		
20	4054261	1	0		

The image to the right shows 4054262 (rev 0) when viewed with an effectivity date of 7/1/2013.

Position	Number	Revis...	Versi...	E	Name
10	4054259	1	0		
10	4054260	2	0		
20	4054263	0	0		

Issues with Effectivity Dates

Currently we don't have a method of changing an effectivity date if the engineer decides that it will not be part of the future production change (or it needs to move out to a later production date). We also have had some issues with revising a part with the same future effectivity date. Currently we have an open SSR for this, but have programmed around it in the meantime.

Opportunities

Classifications - We would like the ability to classify items in order to be able to search by various attributes in the system.

ECP Enhancements - We would like to look at the ECP enhancements that have recently been made by xPLM. Also improvements to saving single level structure and family table improvements.

Graphical Browser - We haven't done any customizations and would like to learn what it takes to have more control over the browser window.

Agile e6.1.2 Upgrade - We will be looking at upgrading to Agile e6.1.2 (or the next available revision) sometime in 2013.

Java Client Speed - Would like to see speed improvements with the Java Client compared to the old Windows Client. (especially from remote locations)

Effectivity Dates - Need ability to change the effectivity date in case something changes for the rollout or it gets pushed out to a different release date.

Contact address:

Rusty Mellmer

NACCO Materials Handling Group, Inc
4000 NE Blue Lake Road
Fairview, OR 97024

Phone: (503) 721-6005
Fax: (503) 721-6200
Email: rusty.mellmer@nmhg.com