ISSN 2320 - 2602
Volume 2, No.9, August 2013
International Journal of Advances in Computer Science and Technology
Available Online at http://warse.org/pdfs/2013/ijacst05292013.pdf

ECLIPSE PLUG-IN BASED TEAMCENTER CUSTOMIZATION

G.Sreenivasulu
Master of Technology (Computer Science Department),
DRK Institute of Science and Technology, Hyderabad, India

ABSTRACT

This paper defines about how to develop a user defined customization in PLM. Depending upon the company's requirements they can choose the different PLM which are existing in the market. This specifying about Teamcenter because it is having more features than the other PLM's which are in the market. This is focusing on Rich client customization.

Teamcenter has more advantages than other PLM's by using this we can integrate this PLM with Different ERP's existing in the market. If we want to done customization in Teamcenter initially we have to develop that in Eclipse later we have to export that to Teamcenter that can be used by the end users. Here how to implementation is also explained.

Key words: Eclipse-plug-in, PLM, Teamcenter

1. INTRODUCTION

Product Lifecycle Management (PLM) is the process of managing the entire lifecycle of a product from its conception through design and manufacture to service and disposal.

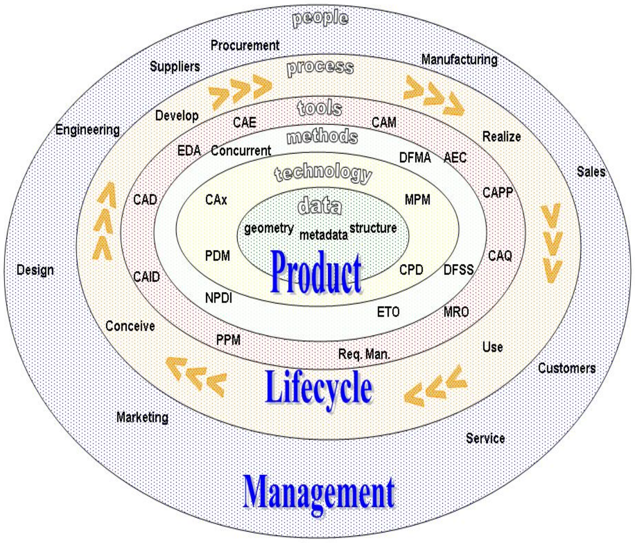


Figure 1: PLM Life Cycle

1.1. Major Players in Product Development Management Market

In Market major players is UGS Team Center Enterprise, Teamcenter Engineering, Dassault Systems Enovia LCA, SmarTeam, VPM, eMatrix, PTC Windchill

2. TEAMCENTER

MyTeamcenter is the workspace you use to manage your product information. MyTeamcenter serves as the main access point for many commonly used functions, such as object creation, check in /checkout, task management, and query execution. We can come across many application of Teamcenter in our day-to-day life some of them are Watch, Cycle, Automobiles, and Defense.

2.1. Teamcenter Architecture Overview

In Teamcenter there are two types of architecture those are Two-tier Architecture, Four-tier Architecture.

2.1.1. Two-tier architecture

In Two-tier architecture the user can develop only Rich client customization. The server is working in his local PC. And the user can store the data in the database in the Local PC.

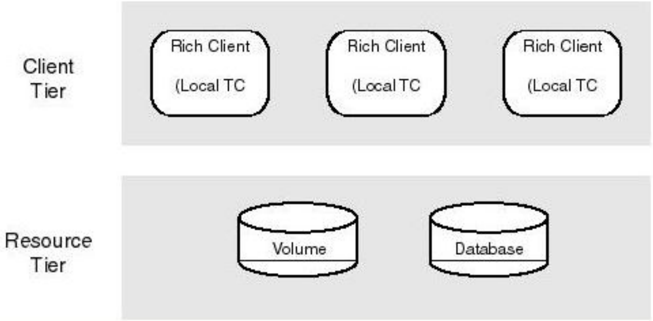


Figure 2: Two-tier Architecture

2.1.2. Four-tier architecture

In Four-tier architecture the developer can develop Rich, Thin and WebDav client. The server is in another location and the developer is making a request to the server through webtier. The webtier is responsible to connect with client-tier and the enterprise tier. If any data has to stored the developer can stored in the database.

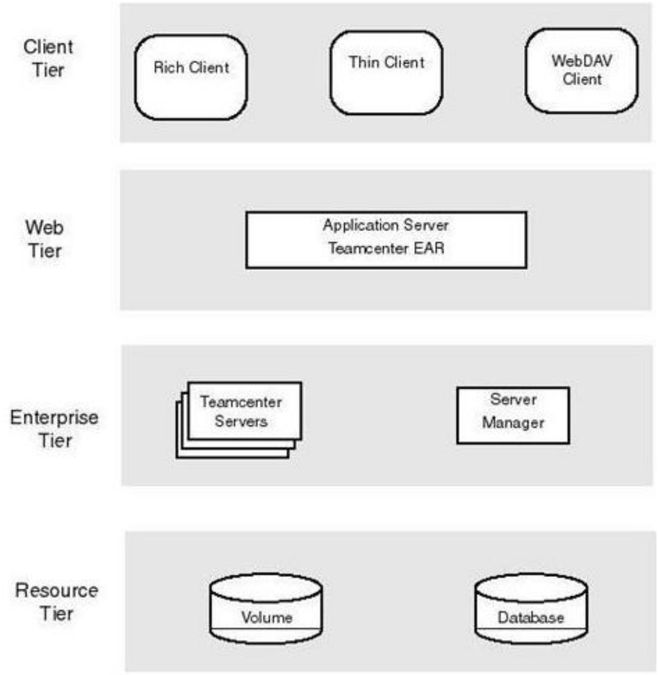


Figure 3 Four Tier Architecture

3. TEAMCENTER CUSTOMIZATION WITH ECLIPSE PLUG-IN

3.1. Need of Teamcenter Customization

Any changes that made out of the box are called customization

Teamcenter is collaborative product data management tool to Help organize all process necessary to design, manufacture, and support a product throughout its lifecycle. The part of Teamcenter power is its adaptability to each company's work Processes and user interface preferences another powerful Teamcenter tool is the ability to integrate data from other applications. You can customize Teamcenter so its meets your Company product life cycle management needs and your users working preferences.

3.2. Eclipse Plug-in

Plug-In is a software component which is having the additional features of the existing software. To Start Plug-In Development in Eclipse initially we need to understand the command Framework.

3.2.1. Need of Command framework for eclipse plug-in

Command framework includes commands, menus, handlers, and category. For example if we take 'File' Menu under this we observe 'n' number of commands if we click on 'New' it will display the new page it is doing some action The code is written in Handler and Category is under one menu to append 'n' number of commands.

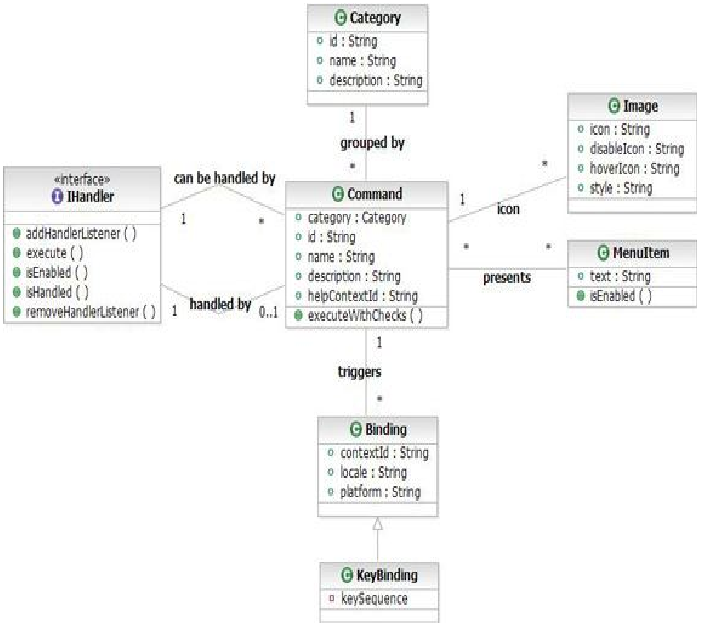


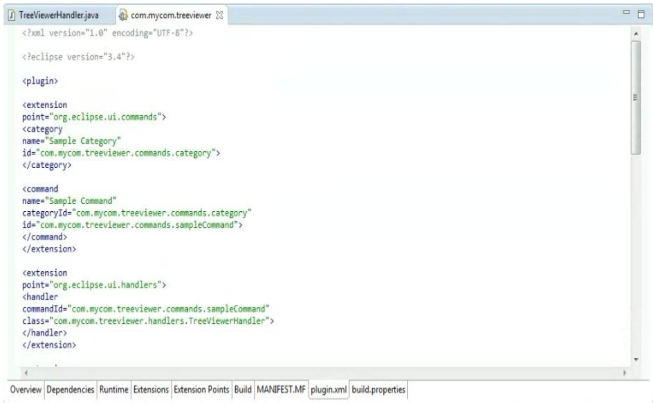
Figure 4: Command framework

3.3. Eclipse Plug-In Development

Initially we have to download the eclipse and install in the user system. After that he have to install a plug-in name called Plug-in Development tool to the eclipse to do customization in team center. After that we have to develop a project and we have to specify the title for the project and proceed to next and it will ask us to develop a rich client or not and proceed to next and it will ask us to choose the template and proceed to next and click on Finish. It will create a project in the project Navigation panel. The user can write the code in the created project and he have to specify the configuration file. The sample plug-in.xml file is shown below

```
<plugin>
<extension point="org.eclipse.ui.commands"><category
name="Sample Category"
id="com.mycom.treeviewer.commands.category">
</category>
<command name="Sample Command"
categoryId="com.mycom.treeviewer.commands.category"
```

```
id="com.mycom.treeviewer.commands.sampleCommand">
</command>
</extension>
</plugin>
```



```
<?xml version="1.0" encoding="UTF-8"?>

<?eclipse version="3.4"?>

<plugin>

<extension
point="org.eclipse.ui.commands">
<category
name="Sample Category"
id="com.mycom.treeviewer.commands.category">
</category>

<command
name="Sample Command"
categoryId="com.mycom.treeviewer.commands.category"
id="com.mycom.treeviewer.commands.sampleCommand">
</command>
</extension>

<extension
point="org.eclipse.ui.handlers">
<handler
commandId="com.mycom.treeviewer.commands.sampleCommand"
class="com.mycom.treeviewer.handlers.TreeViewerHandler">
</handler>
</extension>
```

Figure 5: plug-in.xml

4. Research Work of Exporting Eclipse Plug-in into Teamcenter

If we want to use the plug-in that is developed in eclipse we need to export that plug-in to teamcenter by selecting the application and give right click on it and select the export and then click on the deploy as fragments and click on next and it will ask us to specify the directory to where the plug-in as to export and proceed to next. It will export that plug-in to teamcenter directory. After that he can see that is plug-in is deployed in the directory or not. After we have to go to teamcenter root directory and select the regeistry.bat double click on it will create all bat to regenerate the files and the user has to remove the cache files of the teamcenter and he have to restart the teamcenter to see the effect of the plug-in.

4.1. Usage of Custom Plug-In Inside Teamcenter

The main of the customization is to build the 'n' number of application which we will come to see in over day-to-day life. With the use of the customization the company's may release they software or products into the market to get more profit and reduce the cost for developing the product.

In this application the end user will see the Directory List whatever the Drive he has given he can see all the Directories, subdirectories, files, he can see that in Tree Structure. For example if the users want to know the file where it is located. He can see the Tree Structure and identify the file to which Directory, subdirectory the file is located. This saves the user's time and he can continue on his work.

4.2. Accessing of Custom Plug-in Functionality from Teamcenter

When we click on the Tree Viewer menu item it will display a sub-menu and click on it will perform action and will display the result.

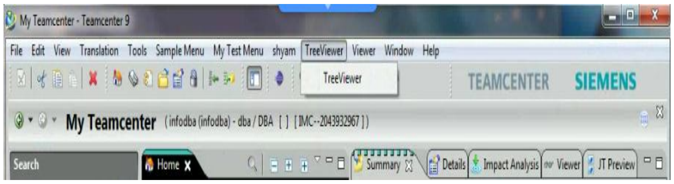
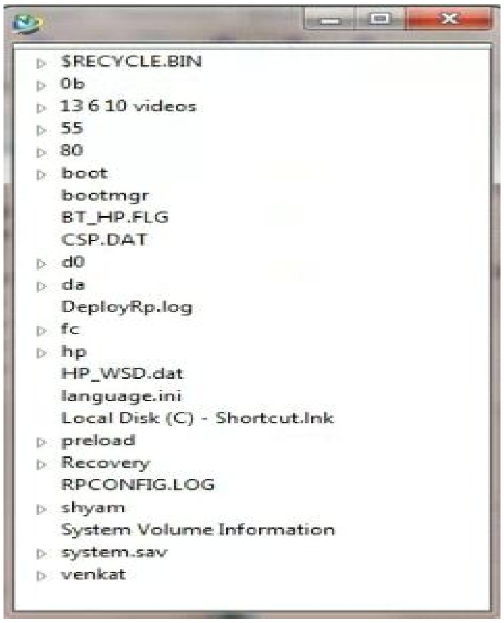


Figure 6: Teamcenter Home page

4.2.1. Final view of custom plug-in commands in teamcenter

When we press on the command the action is going to perform



```
▷ $RECYCLE.BIN
▷ 0b
▷ 13 6 10 videos
▷ 55
▷ 80
▷ boot
  bootmgr
  BT_HP.FLG
  CSP.DAT
▷ d0
▷ da
  DeployRp.log
▷ fc
▷ hp
  HP_WSD.dat
  language.ini
  Local Disk (C) - Shortcut.lnk
▷ preload
▷ Recovery
  RPCONFIG.LOG
▷ shyam
  System Volume Information
▷ system.sav
▷ venkat
```

Figure 7: Final Result of Application

CONCLUSION

In future if the users want to add more capabilities to the existing plug-in he can add that to the existing plug-in. By observing this we can create different views for different commands to the existing menu and we can add sub commands to the existing command that can be give more flexibility to Team center.

REFERENCES

1. Client_Customization_programmers_guide.pdf
2. Teamcenter2007_Architecture.ppt
3. PLM Fundamentals.ppt
4. Configuring and adding menu items in Eclipse V3.3