Implementing Automatic Batch Data Import through Secondary Development of Teamcenter

Gangyu Wu[a], Xianghui Zhan[b, *] Xiaoda Li[c] and Xiaofeng Liu[d]

Zhuhai College of Jilin University, Zhuhai 519041, China

[a]2997981905@qq.com, [b]zhanxh_1215@163.com, [c]liliabc0185@sina.com, [d]996364325@qq.com

*Corresponding author

Keywords: PLM, Secondary Development, Software Engineering, Batch Data Import

Abstract: When PLM projects implementation needs to import large quantities of existing data into the Teamcenter system, it is time-consuming to operate according to the OOTB function (Out of the Box, Teamcenter system original functional modules). In this paper, the customized development method and mechanism of the Teamcenter system are analyzed, automatic batch data import has been developed through custom development of rich clients. The automatic import classification function enables the enterprise to quickly create the Item object in the Teamcenter system and complete the classification by loading the existing item codes, item attribute, classification library number and classification library attribute code on the xlsx file. This function is beneficial to improve the integration efficiency of the existing data of the enterprise and the Teamcenter system.

1. Introduction

With the increasing demand for unified information management of technical data generated during the entire life cycle of a product, Product Lifecycle Management (PLM) came into being. When many companies implement PLM systems, there are already enterprise item management information systems such as ERP systems and a large amount of product data. In order to unify the data management source, these item information such as item number, item name, item attributes, etc. often need to be imported into the PLM system for unified management. To avoid problems such as multiple codes and difficult item queries between different enterprise data management systems.

According to the OOTB function of the Teamcenter system, when the item information is imported, only one Item object can be created at a time. When the number of existing items is large and the types are complicated, the data migration of the items will take a lot of time and effort. Customized development of automatic batch import of items and completion of classification functions to improve the efficiency of PLM project implementation.

The Teamcenter system has a full-featured classification library module, which is convenient for enterprises to classify and quickly find imported items in the Teamcenter system according to classification, attributes and other elements, and improve the reuse rate and calling efficiency of the existing items of the enterprise. Just like the OOTB function of Teamcenter to create item objects, the OOTB function can only complete the classification configuration of one item object at a time. When a large amount of data needs to be classified and configured, there are still problems that are cumbersome and time consuming.

2. Teamcenter system architecture

The Teamcenter system provides developers with a rich secondary development method that can be customized to meet customer needs through interface functions. The Teamcenter system provides four secondary development methods to interact with external programs [1]: RC, ITK, SOA, and Generic Shell. RC (Rich Client Development) is used to change the rich client interface of the Teamcenter system, and trigger the development of programs written in JAVA through a customized system interface [2]. ITK (Integration Toolbox) is used to develop batch programs running on the server using C language. Teamcenter server side provides a wealth of configuration tools, you should try to take advantage of the existing features of the server, reduce the amount of development and avoid the discarding of the program due to system upgrades [3]. Based on the above considerations, this paper focuses on the secondary development of the rich client.

Since the fat client of the Teamcenter system is generated based on the Eclipse plug-in project, the plug-in function can be easily added by using the Eclipse platform, so the function is implemented by the Eclipse plug-in development project and the JAVA language. The system customization development process is divided into: (1). Development tools and environment configuration. (2). Add main application, create plug-in project, add plug-in package, view file, etc. (3). Write menu and response class, write Plugin. Xml file according to Eclipse standard, by writing Command, Menu ContriButions, Handlers add the AbstractHandler subclass of custom interface content and response. (4). Customization can be triggered by a custom menu or toolbar on the interface. [4]

3. Functional overall design

This function needs to complete the large-scale item object creation, attribute assignment and classification in the system with simple operation, instead of the original cumbersome operation. Create a new item type and its attributes in BMIDE. The attribute type is required to be a String character. The classification library framework is built by the system OOTB function, and the classification library ID is recorded (the classification library ID is the identification document of the system identification library), and the classification library ID acquisition location is as shown in Fig. 1 . Sort the item code, item name, data type, and the classification ID of the classification library(if you do not need to classify, fill in XXX here) to be added to the item object that needs to be added to the TEAMCENTER system in the form of xlsx files, The item attribute value, the classification library dictionary attribute value, and arranged in order, as shown in Fig. 2. The first four columns of the first row must be arranged in the order of item code, item name, data type, and classification library classification ID. Each column following the first four columns of the first row represents a item attribute or a classification library dictionary attribute.In order to distinguish between the item attribute and the classification library dictionary attribute, the contents of the column after the first four columns of the first line are composed as follows: the attribute display name plus the identifier "+" or "-", and the "+" followed by the object attribute system internal code name, "-" followed by the attribute ID of the dictionary attribute corresponding to the selected category library in the fourth column. Starting from the second line, the information filled in each line includes all the contents of a item object to be entered into the Teamcenter system: item ID, item name, item type, classification library ID, unlimited number of item attributes, and unlimited number of items. Classification library dictionary attributes. If a line item does not have a value corresponding to the attribute filled in the first line, you do not need to fill it in. When a user imports an object without performing a sorting operation, the classification will not be performed when XXX is filled in.

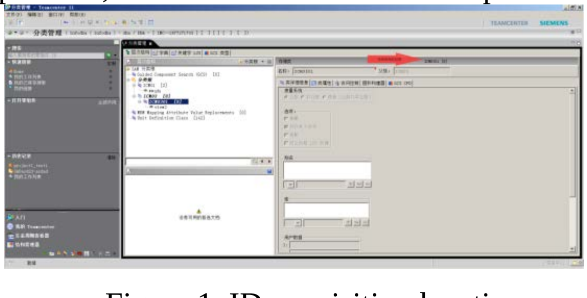

Figure 1. ID acquisition location

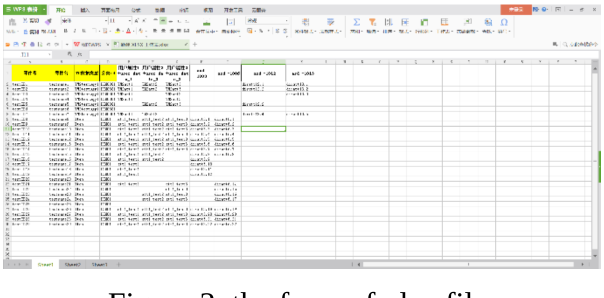

Figure 2. the form of xlsx files

First, the program determines whether the current mouse selected position is a folder for creating a new object. If not, the warning is popped up, the program ends, and the xlsx file is read, and the first row of data is recorded in each column. The corresponding elements in the system, starting from the second line, read the information of the object that needs to be added to the Teamcenter system line by line, read a line of information to process the information of the line, and then read the next line of information until the entire file is processed, and the processing is completed. Starting from the second line, after the program enters the line information, it first determines whether the item code of the object already exists. If it exists, skips the step of creating the object and directly performs the classification operation. If the item code is not present in the Teamcenter system, a new item object is created in the currently selected position of the Teamcenter system by reading the item code, item name, data type, and corresponding value of each attribute of the first line, and then according to whether the item is The classification ID has been added to the classification library and read to determine whether the classification function is enabled. If the item is not added to the classification library and the classification ID is not XXX, the classification function will be enabled. The value of each dictionary attribute in each type of library in the first line is used to classify the library attribute value, and the corresponding object is added in the classification library.

4. Key technology research

This function is developed in the form of Eclipse plug-in project. The program is added as a plug-in to the rich client, all using JAVA language, which determines the current location attribute, creates the object and assigns its attributes, adds the existing objects of the system to the classification library and is the dictionary attribute. Assigning part of the content is a key technique for this feature.

Determine if the current location is a folder. AbstractAIFUIApplication adressClass = AIFUtility.getCurrentApplication();//Get the currently selected location file type

Final InterfaceAIFComponent adrType = adressClass.getTargetComponent();// Determine whether the current location object is a folder

Traversing the xlsx file. The xlsx file is read from the row to the column order with the xlsx file address as a parameter. Create a key-value pair dictionary consisting of a key-value pair dictionary and a classification attribute inner code value and a classification attribute value, which are composed of attribute inner code and attribute value, and record the file content.

FileInputStream stream1 = new FileInputStream(filePath);// Create an input stream

XSSFWorkbook workbook1 = new XSSFWorkbook(stream1);……// Traverse all rows of a particular page

for (int i = 1; i <rowCount+1; i++) {XSSFRow row01 = sheet.getRow(i);// Get current line content

……

}}

Get the attribute and read the corresponding item attribute inner code or classification dictionary attribute ID code through the "+" "-" mark. The key-value pair relationship of the attribute and attribute value is stored in a dictionary.

if (colvaluetemp.contains("+")) {String propname = colvaluetemp.substring (colvaluetemp. indexOf("+") + 1);

Create an item and assign its properties. After inputting the itemid of the object, the item name, the item type, and the value of each attribute, first determine whether the object ID read by the xlsx file is already present in the Teamcenter system. If the object does not exist, the object needs to be created.

mNewItem = TCUtil.newItem(session, item_id, name, mItemType, folder);

Assign a value to the version form of the newly created object:

masterForm.setProperty(key, value);

Add a system existing object to the classification library **and assign a value to the classification library dictionary attribute.** Determine whether the item has been classified and whether the classification ID input value is XXX, and determine whether the object UID is abnormal.

```
if (!mNewItem.getProperty("ics_classified").equals("YES")&&ICMID!="XXX") {
try {
String Uid=item.getUid();//get Uid
TCClassificationService inclass = session.getClassificationService();// get classificationservice
inclass_icm = inclass.newICSApplicationObject("ICM");// Define the ics class
int s = inclass_icm.searchById(id, Uid);
……
inclass_icm.create(id, Uid);
s = inclass_icm.searchById(id, Uid); // get newview, set status
```

At this point, the creation and classification of individual objects has been implemented. By traversing the rows and pages below the file, the automatic import and classification of objects on the entire xlsm table can be completed.

5. Function realization

The function is implemented on the Teamcenter system. The system is WindServer2012, the JDK is 1.7 version, and the underlying database is Oracle11g. You can add custom plug-ins to the system through the plug-in development project of Eclipse4.8.0 platform, and add custom menus in the Teamcenter interface as shown Fig.3.

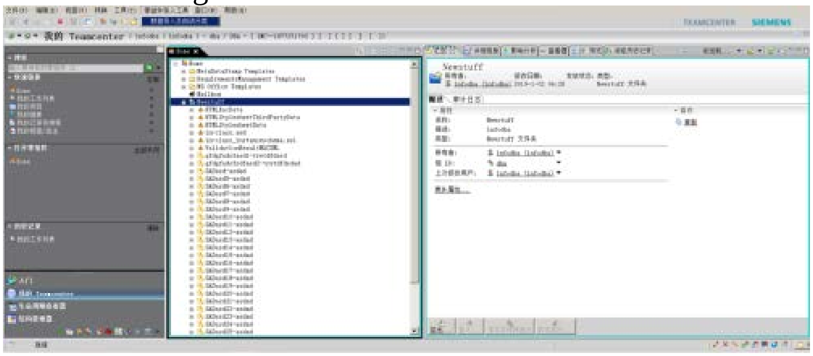

Figure 3. Iadding custom menus

Select the object of a folder class in the Teamcenter system, press the "Data Import and Auto Classification" menu button, the user only needs to select the xlsx file location, the item object creation can be completed automatically and the automatic classification operation can be selectively performed. The program will automatically traverse and record the xlsx content. Use the recorded data as a parameter or intermediate quantity, input it into the JAVA method provided by the Teamcenter system, complete the corresponding item creation, and select whether to perform the classification work according to the requirements. The program test results are in line with expectations, which can greatly simplify the original complicated and repeated operations, and can effectively improve the data interaction efficiency between the enterprise and the PLM system.

6. Summary

This paper studies the automatic creation of item objects and classification functions. By reading the xlsx file loading information, based on the Eclipse platform secondary development technology, using the classes and methods provided by the Teamcenter system, the item objects are automatically created according to their contents. Classification function. At present, this function has been successfully applied in the implementation process of this PLM project, which satisfies the requirement of importing existing data into the Teamcenter system in large quantities through automated methods, and solves the problem of using the OOTB function to realize the complicated operation of the above requirements, which is time consuming and labor intensive. Improve the implementation progress of the PLM project.

Acknowledgements

It is highly appreciated that Siemens PLM software and UGITC provided the NX software and assist. This work is supported by a grant from the JLUZH key disciplines of machinery.

References

[1] GuanHua Chen. An Application Study of PDM System Implementation and Secondary Development Technology [D]. CENTRAL SOUTH UNIVERSITY, 2011.

[2] WANG Lei, LI Yan-jun, CAO Yu-yuan, ZHAO Su-yang, FU Li-xia. Development of Type Certificate Information Management System Based on PDM [J]. Aeronautical Computing Technique, 2016, 46 (01): 103-106+110.

[3] SiDa Wu, ChuanHong Zhou. Implementation of the PSE Product Structrue Acquisition and the Automatic Sorting Function in the Teamcenter Secondary Development [J]. Computer Engineering and Science, 2008 (08): 150-152.

[4] Yong Pan.Research on Development of BOM based on Teamcenter [J]. Science & Technology Vision, 2017 (04): 297-298+301.