

# Research and Application of DELMIA Secondary Development

Chao WU

Hezhou University, Hezhou, CHINA

**Abstract:** Secondary development is an effective means of customization and specialization. Two DELMIA secondary development methods and their customize toolbar methods have been introduced in this paper. In the last part, an example of the secondary development based on Automation method of DELMIA is presented.

**Keywords:** DELMIA; Secondary Development; Customization; Automation

## 1. Introduction

With the development of computer technology, kinds of Computer-aided design software are widely used in industries. These softwares solve the traditional design shortcomings of time-consuming, difficult to modify and so on, to a great extent, save the production time, shorten the development cycle and improve work efficiency. French Dassault Company launched DELMIA as powerful digital manufacture process environment solutions software which is widely used in the digital product manufacturing of aeronautics, astronautics, motorcar and so on. For better customization and extend the service, DELMIA software provides the secondary development interface which contains a series of development tools to help users to implement interface customization, macro command operation, computer-aided design, etc.

Secondary development can focus on user's demand for software to personalized and specialized development, which has important role in improving the work efficiency and product quality, shortening the product development cycle and saving research cost. Nowadays, domestic research on DELMIA secondary development is still in the initial period, and is lack of relevant material, this paper discusses DELMIA secondary development technology and hope to play a valuable role for readers.

## 2. Secondary Development Technology of DELMIA

### 2.1. Two Secondary Development Methods

As a large engineering application software, DELMIA provides many open performances which allow user to customize secondary development conveniently. Two communication methods have been provided by Dassault Company for CATIA and DELMIA: In\_Process Application and Out\_Process Application<sup>[1]</sup>. It is simple to operate In\_Process Application because it based on the macro command which is operated in DELMIA internal, but when the macro command is running, DELMIA is inac-

tive, so it can't accept external data, and the function is limited to be used in simple and repeatable operation process. Out\_Process Application can be used to develop from simple to complex secondary development work with the function of integrated, interactive and user feature definitive, which is the most common secondary development method. Out\_Process Application provides two secondary development interfaces: one is automation object-oriented programming (V5 Automation), which uses Visual Basic programming language to expand DELMIA function; the other is Component Application Architecture (CAA), which uses C++ programming language to expand DELMIA function<sup>[2]</sup>.

CAA is an optimal and efficient method, while it is difficult for abecedarians, learning and programming curve is long. Automation function is limited, but it is easy to grip. Therefore it is propose that development work should use Automation as much as possible, while the deficiency function uses CAA<sup>[3]</sup>.

### 2.2. Flow of Automation Secondary Development

Automation technology, which is based on COM (Component Object Model) and developed from OLE (Object Linking and Embedding) technology, can be carry out in interpret environment, and language-independent and transparent process are its characteristics. The core technology allows one application to operate another one, and it do not access and process data directly, rather, through opening objects and attributes, it uses object methods and properties to achieve, set and process data. Automation contains proprietary constant data which is defined by DELMIA, provides many interface objects and data types. Constant data can be cited directly in communication, and there is no need to define.

Because Automation development method is simple and easy to be realized and modified, so it is becomes a main trend by using the development tool VB, Automation API and Out\_Process Application to develop program. Since DELMIA development libraries is provided by Dassault Company, secondary development process is consistent

with CATIA, which allows to set a specific development according to particular purpose after accessing application object. The developed flow is shown as Fig. 1<sup>[4]</sup>.

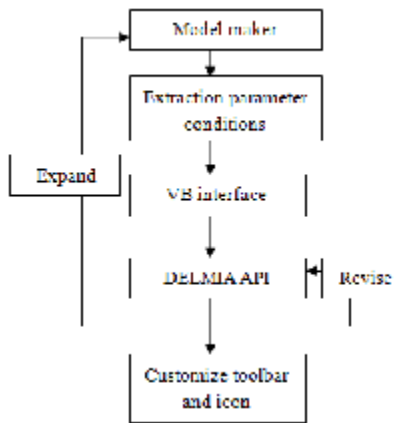


Figure 1. Develop flow of Automation.

it, in this way the program “Nan.exe” will be executed when the macro command runs.

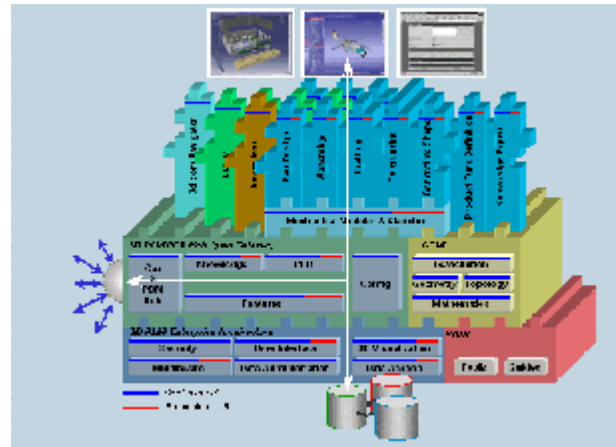


Figure 2. CAA architecture.

### 2.3. Flow of CAA Secondary Development

CAA is component-based platform for Dassault Systemes products to extend and customize development. CAA is tightly combined with DELIA, and makes it very beneficial for integrate and usage<sup>[5]</sup>. CAA can be achieved by RADE (rapid application development environment) and API. By using VC++ as the workbench, RADE tool is completely integrated in the VC++ environment, and provides a CAA framework compiler. API provides methods, tools and interfaces for object manipulation, so user manipulations are able to communicate with DELMIA in VC++ programming environment.

CAA uses object-oriented programming language, and its development process can be seen as a combination an extension of object components. CAA uses COM and OLE technology. The architecture of CAA is shown in Fig. 2.

### 2.4. Customization of Two Toolbar Methods

Generally, secondary development program can exist independently from software interface that users need to find the executable file for running. If users place an icon on toolbar for compiling and running the program by customizing the toolbar, then users can directly use them when they want to. It can save your times, eliminate the interference of using software development with multi-interface and multi-platform. Following are customization toolbar methods of *Automation* and CAA.

1) *For Automation*: It is simple to customize the toolbar in it, which can be implemented without additional code in program. The macro command of DELMIA can be used flexibly.

a) Inserting the command “DELMIA.SystemService.ExecuteBackgroundProcessus “D:\Nan.exe”” into the macro command editor and save

b) Locating the macro by use the “tools”→“customize” option and haul it on the position your wanted, and we also customize icon for macro by properties.

2) *For CAA*<sup>[6]</sup>: CAA customization toolbar is more complex, and used NewAccess (), SetAccessChild () and SetAccessNext () function mainly.

a) NewAccess () function can be seen as a DELMIA workbench entrance bag. Using NewAccess macro can create this entrance. “NewAccess(CATCmdContainer,pCAAfrTB1EltTlb,C AAfrTB1EltTlb)” can creates toolbar which named “pCAAfrTB1EltTlb”.

b) SetAccessChild () function can connects entrance created by NewAccess () function. “SetAccessChild (pCAAfrGeoCreationWkb, pCAAfrTB1EltTlb)” can put toolbar into workbench named “pCAAfrGeoCreationWkb”.

## 3. Example

In this paper, the whole flow of DELMIA secondary development is shown by automatic welding gun and spot assemble program which is developed through *Automation* technology.

### 3.1. Program Development

This program automatically creates the selected welding gun on the corresponding spot customized by users, and it is convenient to adjust the welding gun position and posture in the assembly process and crash analysis for welding gun. DLEMIA and the corresponding parts files and product documents should be active when running the program.

Users use the GetObject () function to link to DELMIA, and using ActiveDocument () method to get the current document. Below is the code:

```
' Achieve DELMIA API
Set DELMIA = GetObject(, "DELMIA.Application")
.....
' Achieve active product document
Set nProductDoc = DELMIA.ActiveDocument
' Achieve active part document
Set nPartDoc = DELMIA.ActiveDocument
.....
```

Users ply AddComponentsFromFiles method to join the customized welding gun in the product. Below is the code:

```
.....
Set nRootProduct = nProductDoc.Product
Set nRootChildren = nRootProduct.Products
nRootChildren.AddComponentsFromFiles CompList, "*"
.....
```

The Selection object contains the features the end user selected, usually with the mouse, and which candidates are as subjects for the next action. Through the Selection.Count2 () property, user can have the count of selected object, as spot number in this program, and user can use Selection.Clear () method to reset it after it is selected.

GetMatrix() function achieves spot coordinate, obtains drawing components and allows user to define the scale, position and reverse it at the same time. Below is the code:

```
.....
Dim DELMIASelection
Set DELMIASelection = DEL-
MIA.ActiveDocument.PPRDocument.Selection
.....
MCount = DELMIASelection.Count2()
DELMIASelection. Clear()
.....
Function getMatrix() As Matrix
Dim matrix As GetMatrix
Set matrix = GetMatrix
.....
```

The point's coordinates and direction selected by user are put into the array, and CreateReferenceFromObject () method adds the reference elements to make welding gun head orientate conveniently. When a user click on "Produce" button, AddComponent () function creates welding gun on the point, then the added position and direction constraints to assemble welding gun with corresponding spot. Below is the code:

```
' Create reference line
Set refPoint =
nPart.CreateReferenceFromObject(nPoint)
Set NewLine = nHSF.AddNewLinePtDir(nPoint,
Array, 0, 10, False)
nHBody.AppendHybridShape NewLine
.....
```

```
' For welding gun component,instantiation
Set refGun = NGun.ReferenceProduct
Set nGun = nRootChildren.AddComponent(refGun)
.....
' Create constraint and the hidden
Set nConstraint = nConstraints.AddBiEltCst
(catCstTypeOn, refGun, refMatrixGun)
Set nConstraint = nConstraints. AddBiEltCst (catCstTy-
peOn, refPAxis, refNewLine)
HideShow oLine nConstraint
.....
```

After running, welding gun is created as the same number of spot, and welding gun head and spot docks vertically. User can adjust the attitude and easily to do crash analysis for welding gun and structure.

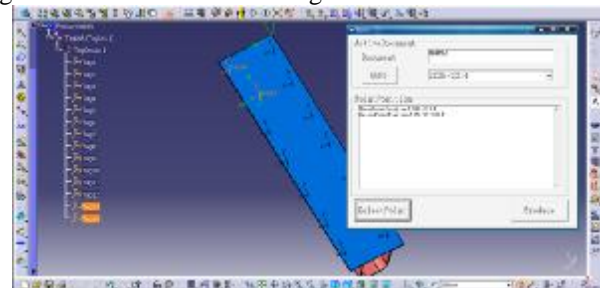
After generating ".exe" files, we customize an icon with the presented methods, as shown in Fig. 3.



**Figure 3. Customize an icon**

**3.2. Results Verification**

User selects welding gun and clicks "Select Point" to choose spot, the program records each spot position. Program interface is shown as Fig. 4.



**Figure 4. Program interface**

After the operation, click "Produce" button, and then program creates welding guns in the selection spots. Fig. 5 shows the result of creating C welding gun and X welding gun in corresponding spots.

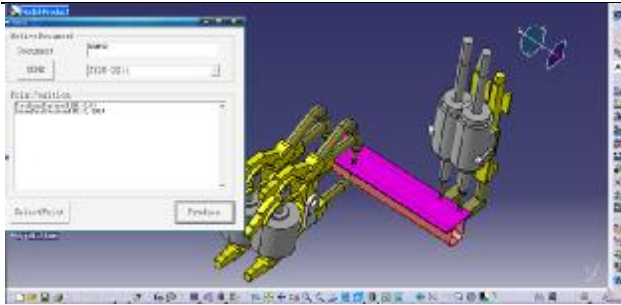


Figure 5. Create welding guns

After assembly, we made crash analysis with the welding gun and structure, and then a possible crash can be found, as shown in Fig. 6.

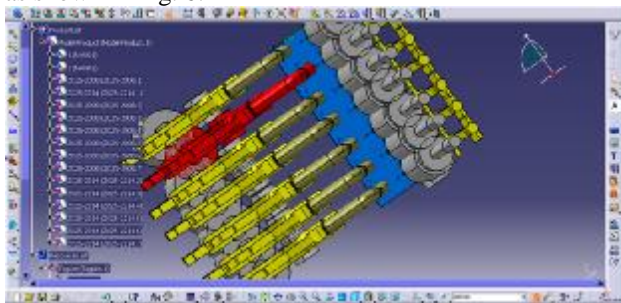


Figure 6. Result and crash analysis

#### 4. Conclusion

As a high-end digital manufacture process environment solutions software, the application field of DELMIA is being enlarged, and its application requirement is also

increase constantly. In order to make better use of DELMIA software resources and to satisfy the user's needs of personalization and specialization, proficient secondary development has become necessary skill for the designer.

This paper discussed common DELMIA secondary development technology of CAA and *Automation*, and their customize toolbar methods. The experiment of using the *Automation* technology to develop an automatic welding gun and spot assemble program shows that the DELMIA secondary development can not only reduce production design time, but also improve production efficiency.

#### References

- [1] G.S. Zhou, W.L. Lu, "Research and Application of CATIA Secondary Development technology," *Machinery Design & Manufacture*, vol. 1, pp. 81–83, January 2010.
- [2] J.F. Wang, "Research on the virtual assembly technology of a light aircraft and the secondary development of DELMIA ergonomics," *Sheng Yang Institute of Aeronautical Engineering*, January 2010.
- [3] X.E. Zhou, M.w.LU, H.X. Zhao, "The Secondary Development of CAA-Based CATIA," *SCIENCE & TECHNOLOGY INFORMATION*, vol. 36, pp. 73–74, 2008.
- [4] T.Hu, L.J.Wu, "Second development technology of CATIA," Peking: Publishing House of Electronics Industry, 2006.
- [5] Z.H.Wang, Q.G.Zhai, "DELMIA secondary development technology research," *Journal of Xi'an Aerotechnical College* vol-28 No-1, pp. 1-3, January 2010.
- [6] Y.X.Dong, P.Xi, "Secondary development of interface based on CATIA system," *AERONAUTICAL MANUFACTURING TECHNOLOGY*, vol.12, pp.83-86, 2006.