

# Intelligent Assembly Technology Based on Standard Parts Feature of CATIA

Baojiang Du<sup>1</sup>, Xiang Wang<sup>1</sup>, Yulin Feng<sup>1</sup>, Dongping Yu<sup>1</sup> & Gang Xu<sup>1</sup>

<sup>1</sup> School of Mechanical Engineering, University of Shanghai for Science and Technology, China

Correspondence: Xiang Wang, University of Shanghai for Science and Technology, 516 Jun Gong Road, Shanghai 200093, China. Tel: 86-881-825-2718. E-mail: 1791745880@qq.com

Received: January 5, 2014 Accepted: February 6, 2014 Online Published: February 21, 2014

doi:10.5539/mas.v8n2p49

URL: <http://dx.doi.org/10.5539/mas.v8n2p49>

## Abstract

In the assembly design of aircraft, ship and automobile industry, it needs to assemble a large number of standard parts. An intelligent assembly technology based on standard parts feature of CATIA is studied in this paper. Through the secondary development of CATIA, it tests the shape feature of assembly base to create the assembly reference and assembly reference in order to realize the intelligent assembly of standard parts and the batch assembly for the standard parts with the same number. This method has been applied in the intelligent assembly systems of an aircraft design. The result proves that it has improved the efficiency of designer's work and reduced the work intensity in the assembly design.

**Keywords:** CATIA, standard parts feature, batch assembly, intelligent assembly

## 1. Introduction

### 1.1 Research Status

CATIA is a computer-aided design (CAD) software developed by France's Dassault aircraft company, which enjoys a high reputation in the field of aerospace, automobile manufacturing, shipbuilding, electronic \ electrical design. Currently CATIA has been the first choice in China where needs a computer-aided design software.

The standards parts library applied in the design of aircraft assembly include fasteners, bearings, piping and electrical parts, containing over 400 kinds of standard parts. Over millions of standard parts are used in the assembly design, so it needs high assembly efficiency.

In order to meet the need of standard parts assembly design based on CATIA, it needs more advanced assembly methods to improve the efficiency. Deqiang Mu proposed an assembly method based on Framework components of the CATIA platform to improve the reuse of a series of products (Deqiang & Ke, 2010). Liuyang Zeng put forward the corresponding algorithms and implementation after studying the assembly constraints based on CATIA / CAA analysis and the interference checking (Liuyang, 2005). Jingzheng Yao completed the virtual assembly of submersible system, analyzing the rationality and feasibility of the entire assembly process by researching the virtual assembly simulation in the environment of CATIA (Jingzheng, Duanfeng, & Jian, 2011). All of these methods about assembly technology just could make the designers to establish the constraints manually and the assembly product but can't update while an error happen in assembly. Therefore these methods can't meet the complicated assembly work and the intelligent assembly during the design of aircraft.

The paper put forward a way to achieve the intelligent assembly of standard parts based on its features (Zhongyi, 2010). Through the secondary development of CATIA, this method extracts the assembly feature, adds the constraint relations automatically and achieves intelligent assembly based on assembly feature (Lin & Mengchun, 2007).

### 1.2 Secondary Development of CATIA

There are a variety of ways to access the CATIA object in the secondary development of CATIA (Hongjun & Yongjin, 2008). The In~Process application has access to CATIA object by VBScript or VBA script. The VBScript or VBA script that is recorded by Macro in CATIA can be modified moderately to meet the requirements of users. The Out~Process application has access to CATIA object by Component Object Model(COM) While CATIA would be of Object Linking and Embedding (OLE) (Ting & Lijun, 2006).

Here we choose VB.Net as the main program language for the secondary development of CATIA. The main theory is to establish an assembly system by Visual Basic.NET (VB.NET) as an Out-Process application to access to CATIA object. And the VB.NET provides a lot of functions for designers.

## 2. Method

### 2.1 Detection of Shape Feature for Assembly Base

The feature generally refers to the shape feature. In fact, the features of standard parts can be divided into shape features, materials features, assembly features, precision features, analysis features and technology management characteristics in the application process of CAD, CAPP and CAM.

#### 2.1.1 Extraction of Shape Feature for Assembly Base

In the modeling process, shape features can be divided into the main shape feature and the secondary shape feature. The main shape feature is about the model geometry and the secondary shape feature is characterized by the details of the main shape feature such as fillets, chamfers and threads (Dassault, 2006).

The shape feature of assembly base is detected mainly by analyzing the selection of the designer on the assembly base such as face, line and point (Sheng, 2010). Detection of shape feature for assembly base is extracted to create assembly reference such as the assembly surface, the assembly axis and the assembly point. The process of detecting the shape feature of assembly base is divided into two parts. One part is that creating the assembly reference and another part is that getting the assembly reference size.

The most standard parts applied to the aircraft assembly design are a variety of bolts, nuts, rivets and other fastenings. Accordingly, the establishment of assembly constraints only needs one assembly axis feature and an assembly face feature. If the selection on assembly base is a cylinder, the path of assembly dependencies tree with the axis feature of model will be extracted to generate assembly axis feature. On the contrary, if the selection is plane, the path of assembly dependencies tree will also be extracted to generate assembly surface feature. Extraction is converted to assembly reference in program.

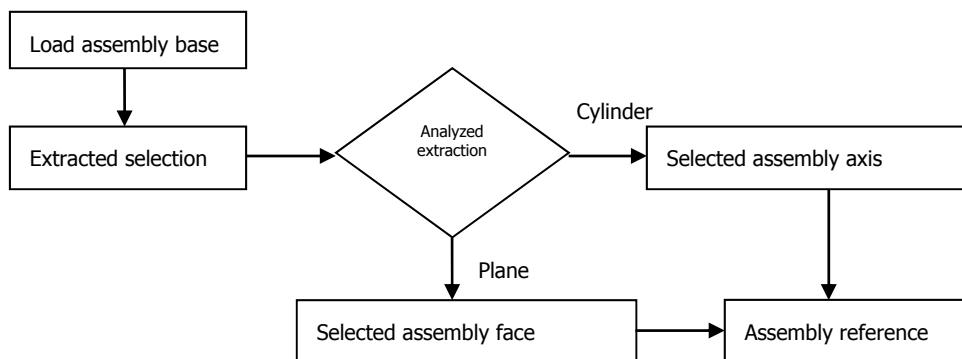


Figure 1. Flow chart of assembly reference produced

#### 2.1.2 Establishment of Assembly Reference

Extraction of shape feature is imported to the database to establish assembly reference. Assembly reference is created by the use of the CreateReferenceFromName function which type is the Reference in VB.NET. The name of assembly reference is made up of assembly product name, assembly base name and the name of shape feature selected in VB.NET program. For example, the code of creating the assembly face is the following:

```

ReferenceFace = Product.CreateReferenceFormName(Product.Name & "/" &
Selection.Item(i).LeafProduct.Name & "/" & Selection.Item(i).Reference.Name)
  
```

The feature matched with standard parts feature is mostly hole feature or axis feature. The assembly reference size can be measured by the selection on the cylinder in the program. The workbench in model structure tree is detected to export the radius of cylinder by the GetMeasurable function. Extractions of shape feature for assembly base make foundation for intelligent assembly and batch assembly.

## 2.2 Establishment of Assembly Constraints

### 2.2.1 Publication of Assembly Features

Assembly feature as a part of the model features includes size features, vowel features, and relation characteristics. Constraints cannot be added to the geometric of standard parts by intelligent assembly design directly, because the geometric of the model that is always built by revolving or extruding is topological element, which is not fit for the intelligent assembly in this paper for the reason that the topological element has no generality in the process of assembly design. Therefore, we should publish the assembly feature before establishing the constraints.

In the modeling process, assembly features of standard parts are published by the use of CATIA publication function, including assembly face, assembly axis and assembly point (Bingfeng, 2009). According to the type of constraint added to assembly feature, the corresponding name is defined for the assembly feature published.

Table 1. Publication name and constraint type of different assembly features

Published Feature	Publication Name	Constraint type
Face	Contact Face	Surface Contact
Face	Coincidence Face	Surface Coincidence
Face	Offset Face	Surface Offset
Line	Contact Axis	Axis Contact
Line	Coincidence Axis	Axis Coincidence
Line	Offset Axis	Axis Offset

As shown in Table 1, Different assembly features correspond to the different Publication names and constraint types.

### 2.2.2 Establishment of Assembly Constraints

In the assembly process, the constraints limit the relative position between assembly base and standard parts. Degree of freedom which is limited by the contact constraint and coincidence constraint enforces the localization of standard parts. Furthermore constraints only are established between the models with the same constraint type of assembly feature. The constraint type of assembly feature of standard parts should be compatible with the assembly base because the constraint type of assembly base has been identified when the assembly reference is established. Therefore, we established the constraints between the assembly base and standard part due to their constraint type in Table 1. The two kinds of constraint programs based on assembly axis feature or assembly face feature are established by the AddBiEltCst function in CATIA secondary development. The main constraint program is the following:

```
constraint = oConstraints.AddBiEltCst (iCstType, iFirstElem, iSecondElem)
```

iCstType ---- the constraint between assembly models

iFirstElem ---- the assembly feature of standard part matched

iSecondElem ---- the assembly reference on assembly base

The constraint in program is expressed by the constant value. For example, in the program the axis coincidence constraint is expressed by the catCstTypeOn which constant value is 2. The catCstTypeSurfContact means surface contact constraint which constant value is 20. Due to different constraints, corresponding constant value of constraint is used in the system program.

Table 2. The different constraints

Constraint expressed in program	Constraint	Constant value
catCstTypeReference	Fix constraint	0
catCstTypeDistance	Offset constraint	1
catCstTypeOn	Coincidence constraint	2
catCstTypeAngle	Angle constraint	6
catCstTypePlanarAngle	Planar Angle constraint	7
catCstTypeParallelism	Parallelism constraint	8
catCstTypePerpendicularity	Perpendicularity constraint	11
catCstTypeSurfContact	Surface Contact constraint	20
catCstTypeLinContact	Line Contact constraint	21
catCstTypeAnnulContact	Annul Contact constraint	25

As shown in Table 2, it records the constraint expressed in program and constant value of different constraint.

The two kinds of codes creating constraint relations between assembly features in the model are the follows:

```
oConstraints = product1.Connections ("CATIAConstraints")
```

```
'establishment of constraint between assembly axis feature, catCstTypeOn = 2
```

```
constraint1 = oConstraints.AddBiEltCst (2, PubAxis, ReferenceAxis)
```

```
'establishment of constraint between assembly face feature, catCstTypeSurfContact = 20
```

```
constraint2 = oConstraints.AddBiEltCst (20, PubFace, ReferenceFace)
```

According to the extracted assembly features of assembly base in assembling process, the assembly system automatically chooses corresponding constraint program to achieve the assembly of standard parts.

### 2.3 Batch Assembly

The assembly system tests the selection information about assembly position of the assembly base to judge whether it is single or batch assembly. The results of testing and assembly position selected are imported into the database. If it is batch assembly, the program retransfers standard parts to the next assembly location and to match the assembly feature with the assembly base, which realizes the batch assembly.

The method of batch assembly utilizes a loop program to read the number of selected features and assembly positions of the assembly base. The Selection function and the Selection.Item(i) function are used to obtain the first i selected feature and assembly position. Finally, the standard parts with the same part number are transferred to the selected position one by one and constraints are established which realizes the batch assembly.

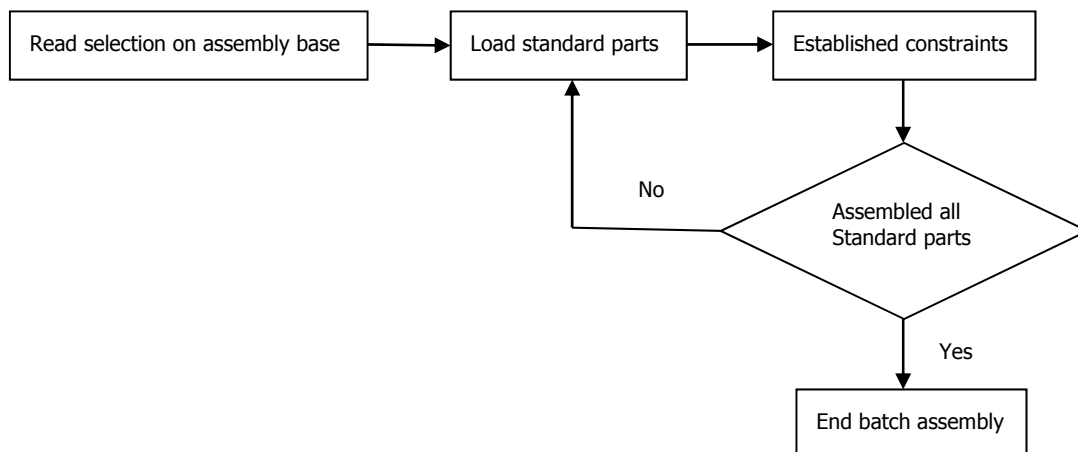


Figure 2. Flow chart of batch assembly

As shown in Figure 2, the function should be written according to this flow chart.

### 3. Results

Using this method, we establish the aviation standard parts library of intelligent assembly system for an aircraft design institution which contains two millions standard parts. Take mounting bolt in the steel plate as an example to validate this method.

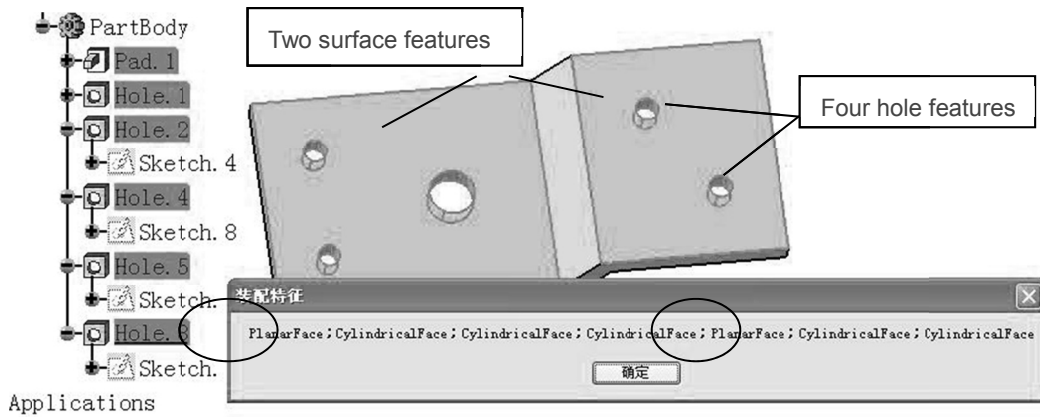


Figure 3. The model of steel plate

As shown in Figure 3, it is the model of steel plate. Furthermore we selected four hole features and two surface features. Due to the selection, intelligent assembly system extracts the assembly information to establish assembly reference.



Figure 4. Bolt NAS6703AHL2 load

As shown in Figure 4, the bolt NAS6703AHL2 is being loaded from the aviation standard parts library of intelligent assembly systems.

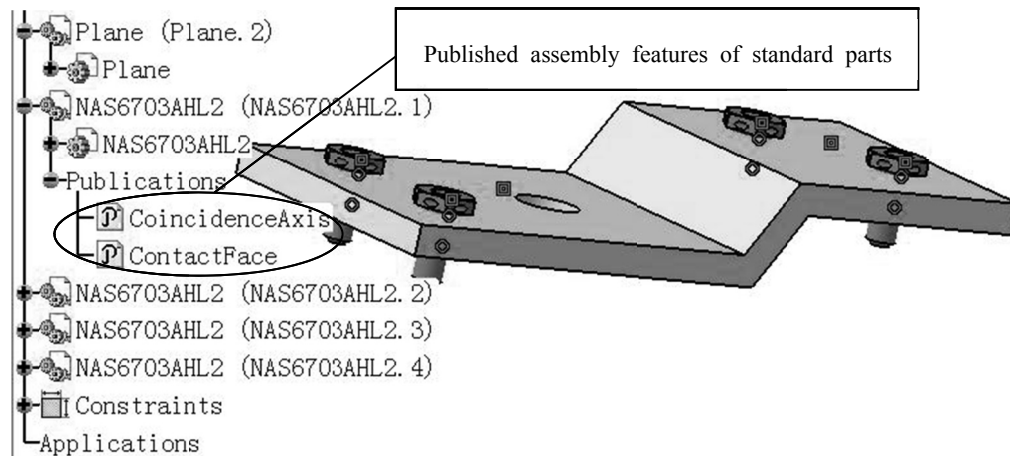


Figure 5. Four bolts assembled

As shown in Figure 5, four bolts are assembled at the same time quickly.

#### 4. Conclusions

Intelligent assembly based on standard parts feature overcomes the trouble of looking for assembly constraints constantly and avoids the error caused by modifying the individual standard parts in assembling process that improves assembly accuracy. It shortens the design cycle and proves that the technology is necessary and feasible. With the development of technology, intelligent assembly will be applied further.

#### Acknowledgements

Thanks Gang Xu for writing assistance, Yulin Feng for providing language help and proof reading the article.

#### References

- Alemanni, M., Destefanis, F., & Vezzetti, E. (2011). Model-based definition design in the product lifecycle management scenario. *International Journal of Advanced Manufacturing Technology*, 11, 1-14. <http://dx.doi.org/10.1007/s00170-010-2699-y>
- Alok, M. (1999). *Generalized Fluid System Simulation Program Version 3.0*. Technical Report, Report No: MG-99-290, Marshall Space Flight Center, November, 1999.
- Bingfeng, H. (2009). Application of Publication in CATIA. *Journal of Hubei Automotive Industrial Institute*, 23, 9-11.
- Curran, R., Collins, R., Poots, G., Edgar, T., Higgins, C., & Butterfield, J. (2008). Digital Lean Manufacture (DLM): A New Management Methodology for Production Operations Integration. In *Collaborative Product and Service Life Cycle Management for a Sustainable World* (pp. 551-571). Springer London. [http://dx.doi.org/10.1007/978-1-84800-972-1\\_52](http://dx.doi.org/10.1007/978-1-84800-972-1_52)
- Dassault Systemes. (2006). CAA V5 for CATIA Foundations.
- Dassault Systemes. (2006). CATIA V5 User's Documentation.
- Deqiang, M., & Ke, W. (2010). Framework Assembly Base on CATIA Platform. *Journal of Changchun University of Technology*, 31, 28-31.
- Dixon, J. R. (1987). Research in designing with features. In *Proceeding IFIP TC Workshop on Intelligent CAD*, 10, 137-148.
- Duffy, A. H. B., & Duffy, S. M. (1996). Sharing the learning activity using intelligent CAD. *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, 10(2), 83-100. <http://dx.doi.org/10.1017/S0890060400001335>
- Gardan, Y., & Minich, C. (1993). Feature-based models for CAD/CAM and their limits. *Computers in industry*, 23(1), 3-13. [http://dx.doi.org/10.1016/0166-3615\(93\)90110-M](http://dx.doi.org/10.1016/0166-3615(93)90110-M)

- Gomes de Sá, A., & Zachmann, G. (1999). Virtual reality as a tool for verification of assembly and maintenance processes. *Computers & Graphics*, 23(3), 389-403. [http://dx.doi.org/10.1016/S0097-8493\(99\)00047-3](http://dx.doi.org/10.1016/S0097-8493(99)00047-3)
- Hongjun, S., & Yongjun, W. (2008). Research on CATIA Second Development Based on CAA. *Journal of Mechanical*, 1, 41-43.
- Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., & Hart, P. (1999). VADE: a virtual assembly design environment. *Computer Graphics and Applications, IEEE*, 19(6), 44-50. <http://dx.doi.org/10.1109/38.799739>
- Jingzheng, Y., Duanfeng, H., & Jian, L. (2011). Virtual assembly simulation based on CATIA system. *Journal of Marine engineering*, 15, 49-52.
- Jung, S., Hoffhenke, M., & Wachsmuth, I. (1997). Virtual assembly with construction Kits.
- Lin, T., & Mengchun, Z. (2007). Research on automatic assembly based on assembly feature modules. *Journal of Mechanical design and manufacturing*, 10, 159-161.
- Liuyang, Z. (2005). *Constraint Relation Analysis and Interference Research of Virtual Assembly Based on CATIA/CAA*. Nanjing University of Aeronautics and Astronautics.
- Sheng, L. (2010). Based on CAD/CAPP integration, the accuracy of the information modeling and extracting. *Journal of CAD/CAM and manufacturing information*, 8, 24-26.
- Zhongyi, M. (2008). The technology of aircraft digital assembly based on MBD. *Journal of Aeronautical Manufacturing Technology*, 18, 42-45.
- Zipori. (2008). *Digital Manufacturing Processes in a Plm Envir- Onment for the Aerospace Industry* (pp. 36-38). McGraw-Hill.

### Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).