Example-based Reasoning for Deformation Design of Typical Aerospace Parts

Liu Ming*
Department of Mechanical Engineering, Shandong Vocational College of Science and Technology, Shandong, China

*Corresponding author: liu_ming@sdvcst.edu.cn

Abstract. To address the tedious process of modelling typical aviation parts, it is difficult to achieve rapid modelling phenomenon. Component Application Architecture, or CAA, is used to develop CATIA. It proposes a deformation design method for typical aeronautical parts based on example reasoning. A library of typical part models is first created, then similar models are retrieved by instance retrieval and matching, followed by retrieval of similar models. The next step is to analyze the target design model and the retrieved 3D model, and then make changes to its features to complete the deformation design. The results show that the deformation design method can quickly and accurately create 3D models, improve design efficiency and quality, and provide an effective guarantee for the processing and manufacturing of typical aeronautical parts.

1. Introduction
The aviation manufacturing industry has recently developed at a rapid pace. Typical parts are the main constituent parts of aircraft fuselage structures and wings. In the field of aviation design, typical parts have complex structures, high design requirements and high machining accuracy, but the traditional 3D modelling method has a lot of repetitive data query and manual input operations. This increases the modelling burden on designers, reduces modelling efficiency and makes it difficult to meet rapid modelling requirements. In this paper, the design of a typical aeronautical part is deformed, and based on the similarity of the structure of the typical part, similar models in the model library are retrieved using example reasoning to realise the variant design based on example reasoning, and finally the model is built quickly. This not only reduces the repetitive operations of typical part design, but also enables the 3D model to be reused, thus improving the annotation of the design.

2. Typical parts for aviation
Typical parts for aviation are parts with similar structure and machining process, in large quantities and reusable with commonality. Aircraft structure mainly consists of fuselage, wing, tail, power unit and take-off device components. Typical parts are mainly found in the wings and fuselage. They mainly include add-on parts, wing beams, sheet metal parts and profiles. This paper describes several typical parts as follows.
2.1. **Tie rods.**
Tie rods are simple in construction and are mainly used for connecting stressed rods inside the wing, where the slots in the side of the tie rod are used to relieve concentrated load stresses. The tie rods can be divided into tie rods 1-1, tie rods 1-2 and tie rods 2-2 depending on the actual need.

2.2. **Sheet metal parts.**
Aircraft sheet metal parts are mainly composed of webs, airfoils and edge strips, which are sheared and stamped to produce sheet metal parts of the same thickness. According to the number of flanges, the sheet metal parts can be divided into two-sided, three-sided and four-sided flanged sheet metal, and according to the existence of openings, they can be divided into two types: openings and non-openings.

2.3. **Long truss profiles.**
It is mainly subjected to shear forces caused by local aerodynamic loads and axial forces caused by the bending moment of the wing. Long joist profiles can be divided into normal long joist profiles and skinned long joist profiles [8].

2.4. **Wing beam.**
It is mainly subjected to longitudinal forces and consists of a web and a rim strip. The edge bars are mainly subjected to tensile and compressive deformations and bending moments. The shape of the wing beams is similar to that of sheet metal parts. They can be divided into malformed wing beams, Z-shaped wing beams, I-beams and hooked wing beams according to the shape of the end sections, and can be further subdivided according to whether the ends of the wing beams are closed, open or notched.

3. **Case-based Reasoning**
Case-based Reasoning [1] is a similarity-based intelligent design that uses existing empirical knowledge to solve real-world problems and applies the relevant knowledge information from the examples to the new case method. Analogical reasoning is built on the essence of the example reasoning method. The designer classifies the already created typical aeronautical parts according to their structural characteristics and stores them in a model library, while storing the model information of the typical parts according to certain rules. When the designer changes the existing model information according to the actual requirements, he can use the example reasoning mechanism to search for the 3D model with the highest similarity to the target example in the typical part model library, and then design a variant of the typical aeronautical part to complete the rapid change of the model.

The structural framework for example-based reasoning is shown in Figure 1.

![Figure 1. Structural framework based on example-based reasoning](image-url)

As shown in Figure 1, the deductive reasoning module mainly consists of a domain rule base and a deductive reasoning machine. Its main function is to decompose the design task, control the solution process and complete the modification of the selected design cases. The inductive reasoning module consists of an inductive reasoning machine and a library of design examples. The principle is to use the library of design examples to store existing design model examples and to match and retrieve the target
examples according to the design requirements. The data library and the model instance library mainly store information such as engineering drawings, 3D models, parameter details and design documents of design instances. Researchers use this information to manage the design rules and constraints in the instance library. The user interface module is mainly for the exchange of information between the system and the model designer. Example reasoning calculates the similarity between the models in the typical model library and the design models based on the geometry and parameters of the models, so as to further retrieve and then modify the geometric parameters on the basis of the existing model example information. The final example model with a high degree of model matching is achieved. The method is easy to maintain and of high efficiency, which improves the efficiency of using the existing experience and avoids new problems arising in the initial stage.

4. Instance retrieval and matching

Instance retrieval is a key part of the process of selecting existing models with high similarity to the design model by indexing them from a library of created model instances. The efficiency and accuracy of instance retrieval directly affects the operation mechanism of the whole variant design module, which needs to compare the similarity of the instances from the perspective of their individual characteristics [2]. The main methods of instance retrieval are the following: nearest neighbor strategy, grey correlation algorithm, inductive guidance strategy, knowledge-based neural index retrieval method and knowledge-guided method [3]. As this paper is based on the idea of human-computer interaction design and the use of instance inference to achieve instance retrieval, the nearest-neighbor strategy approach is chosen to achieve instance retrieval and matching. The method requires the designer to input specific design requirements, and the model information data in the typical model library is matched against the model design requirements to retrieve the most similar example model by comparison. The main design flow of the instance retrieval is shown in Figure 2.

![Figure 2. Example search design flow](image-url)
(1) Determine to retrieve model feature information and search for related information. Analyze the geometric information and parameters of the design model according to the model example design requirements. With reference to this model information, determine whether the required model instance exists in the typical model library. If it does not exist, the design model feature information is re-analyzed and calculated; if it does exist, the similarity of the two model feature information is calculated. The main part of the procedure for this operation step is as follows.

\[
\text{if} \quad \text{CATUnicodeString}(\text{"FFFF")} = \text{CurrentSelect} \quad \text{// Determine if a model instance exists in an existing model library}
\]

\[
\{
\text{CreateUDFBoss()};
\text{CATUnicodeString Temp} = (\ast \text{pListOfInputRole})[3];
\}
\]

(2) Calculation of the similarity of model geometry parameters. By analysing the typical parts in the model library, the similarity is determined using the distance for calculation, which is a simple and feasible process compared to other methods.

If there are two instance models, \(u\) and \(v\), with \(u\) being the target design instance model and \(v\) being a model instance that already exists in the typical model library. Then the similarity \(\text{sim}(u, v)\) is:

\[
\text{sim}(u, v) = 1 - \text{dist}(u, v) = 1 - \frac{\text{dist}(ui, vj)}{\text{maxi} - \text{mini}}
\]

\[\text{dist}(ui, vj) = \frac{|ui - vj|}{\text{maxi} - \text{mini}}\]

\(\text{maxi}\): the maximum value of the \(i\)-th geometrical parameter.

\(\text{mini}\): the minimum value of the \(i\)-th geometrical feature.

\(\text{Dist}\) is equal to 0 if the \(i\)-th geometric feature parameter in \(u\), \(v\) is the same value, otherwise it is equal to 1. Part of the program code is as follows.

\[
\text{CATPathElement PathFirstInstantiate1} = \text{SelectedPathMent[1]}; \quad \text{//Determining the model building blocks}
\]

\[
\text{FilterTheSelectCATPathMent} \quad \text{(FirstPlane, TempFirstPlane1, pIUdfInstantiate)}; \quad \text{// Model feature information matching and discrimination}
\]

\[
\text{TempFirstPlane1} \rightarrow \text{Release}() \quad \text{//Pointer release}
\]

…….

5. Variant design of typical aeronautical components

This refers to the retrieval of an existing model from a typical model library by means of an example search and then its variant design. The variant design refers to the modification of geometric parameters or local features on the basis of existing model features. The essence of this modification is to modify the geometric parameters of the model and add or subtract features without changing the original model design principles. The example-based variant design studied in this paper calls up existing models with a high degree of similarity in a typical part model library, then uses the CATIA V5 super copy function to make a copy of the existing model, and then modifies the model geometry parameters, and locally adds or subtracts model geometry features according to the designer's needs.
The variant design refers to the modification of an existing 3D model by comparing and analyzing the target design model with the existing model in the model library to find the differences between the two models [4], using the Super Copy function in CATIA V5 to copy the original model features according to the new modelling base, and then modifying the model features according to the design requirements. This method essentially decomposes the structure and function of the existing typical part model and then completes the creation of the part model feature mapping relationship. The variant design principle requires the target design model to be compared with and aligned to the model in the model library according to the design requirements, and the corresponding part feature modifications to be made on top of those already created to complete the variant design of a typical aviation part. The working principle is shown in Figure 3. The working principle is to analyse the existing model and the design target, then determine the feature information and constraint relationship of the model, then decompose the structure of the model, complete the transformation of feature parameters and realize the deformation design of the model.

![Variant design flow](image)

**Figure 3.** Variant design flow

6. Example of rudder suspension joint variant design

Based on the design and modelling process of a typical aeronautical part, the principles of parametric model design are used to modify the parameters of the model features based on the premise of example-based reasoning. We can also perform super-copy and Boolean operations on features such as tabs, slots and holes of existing models to add, delete and splice features of the part. The process of variant design is described here using a typical aeronautical part, the rudder suspension joint, as an example. The specific steps are as follows:

1. Obtain the rudder suspension joint part edit file
   
   ```c
   CATDocument * pNewInstanceDoc = spFrameEditor->GetDocument(); // Get current documentation on rudder suspension joints
   CATIContainerOfDocument_var spConODocs = pDoc; // Definition of variables
   ```

2. Search for geometrical characteristics of rudder suspension joints
   
   ```c
   CATLisValCATISpeObject_var * pLisUserFeature = NULL;
   CDiallogState * pInputState = GetInitialState
   pListPowerCopy = pIUfdFactory->GetPowerCopyList(); // Retrieval of feature parameters
   ```

3. Creating a super copy of the rudder suspension joint. The model of the rudder suspension joint has already been created in the model library and we have published its geometric parameters and relationships between parameters in the CATIA structure tree. At this point we open the rudder suspension joint model retrieved based on the example inference, select the feature elements in the
model structure tree and create a super copy. We then select the XY plane and the lug axis in the structure
tree as input copies, and the outer radius of the lug, inner diameter of the lug, lug thickness, slot width
and end thickness as parameter copies. The super copy is created and saved as shown in Figure 4.
(4) Creation of building elements (modelling datum). We need to recreate the modelling datum
according to the structure of the design target model, select the new modelling datum in the human-
machine interface of the variant design module and determine the location of the target model. This is
shown in Figure 5. The modelling datum for the rudder suspension joint will replace the existing
modelling datum for the lug axis, support end face, lug centre face and end face edge line. Some of the
main program code is shown below.
   pListUserFeature = pUdfFactory->GetPowerCopyList();
   rc = pUdfFactory->GetOldInputs(pListOfInput,pListOfInputRole);//Modelling
   benchmarks already available
   rc = pUdfFactory->SetNewInput(1, TempFirstPlane);//Selecting a new modelling baseline
   ……
(5) Instantiated interface queries for 3D model features. Some of the main program code is shown
below.
   rc = pUdfInstantiate->Instantiate (FirstDest1);
   if (FAILED (rc))
   {
      cout << " Error in the Instantiate method" << endl;
      return;
   }
(6) A feature traversal method is used to obtain the list of parameters and the list of elements of the
super copy. Some of the main program code is as follows.
   CATIUdfFactory *pIUdfFactory = NULL;
   rc = pPrtConOnCAAUdfBoss->QueryInterface(IID_CATIUdFactory,(void**)
   &pIUdFactory);// Get the query interface
   pListUserFeature = pUdfFactory->GetPowerCopyList(); // Get the list of elements
   rc = pUdfInstantiate->GetParameters(pListParam,pListParamRole);//Get the list of parameters
   ……
(7) Instantiation of 3D model features and completion of feature changes. Some of the main program
code is shown below.
   pUdfInstantiate->Instantiate ();
   pUdfInstantiate->End instantiate().

   The design of the rudder suspension joint can be deformed by the above mentioned steps, allowing
the rapid creation of the target model. We can also modify the parameters of the 3D model to drive the
geometric parameters during this deformation design process. The rudder suspension joint before and
after deformation is shown in Figure 6.

Figure 4. Comparison of rudder suspension joints before and after variation
7. Conclusion
In this paper, a variant design method based on example-based reasoning was used to change the structural feature parameters of a parametrically created 3D model. This allows the 3D model to be created accurately and quickly, reducing many repetitive modelling steps, saving the designer's time, improving the efficiency of the design of typical aeronautical parts, and having positive implications for the subsequent simulation and processing of the part.

References
[1] Liu Teng. Research on parametric design technology of typical aerospace parts based on knowledge engineering [D]. [Master's thesis]. Shenyang: Shenyang University of Technology, 2016.
[2] Wang Lei. Research and design of data retrieval algorithm based on instance inference [D]. [Master's thesis]. Shenyang: Shenyang University of Technology, 2013.
[3] Chen Jun. Example-based research on key technologies in automotive parts product development [J]. Journal of Hubei Institute of Technology, 2013, 29(1): 32-35.
[4] Liu M, Hao B, Sun Shengli. Variant design of typical aircraft parts based on CATIA technology[J]. Group technology and production modernization, 2016, 33(1): 6-11.