Springer

Journal of Mechanical Science and Technology

Journal of Mechanical Science and Technology 23 (2009) 246~253

www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-008-0928-1

Research on method of constraint conversion in feature-based data exchange between heterogeneous CAD systems †

Wei Sun, Tie-qiang MA* and Yu-jun Huang

School of Mechanical Engineering, Dalian University of Technology, Dalian, 116024, China

(Manuscript Received July 2, 2008; Revised September 10, 2008; Accepted September 26, 2008)

Abstract

Data exchange between heterogeneous CAD systems is the key problem in the field of CAD, and the popular solution to this problem is a feature-based method. Aiming at keeping the consistency of design intent in data exchange between heterogeneous CAD systems, this research focuses on the method of constraint conversion. First, two kinds of basic feature transformation ways are summarized, and this is the foundation of constraint conversion. Second, a novel identification of geometric elements based on the improved geometric certificate is proposed, and the identifier of geometric element is used to recognize the acting object of constraint. Third, the principles of constraint conversion are put forward, and the method of constraint conversion for a single feature is proposed according to the principles. Fourth, an algorithm of constraint conversion for the whole process of CAD data exchange is designed. Finally, the contrast experiment of CAD data exchange is completed, and the result of the experiment shows that the approach mentioned above is effective.

Keywords: Heterogeneous; CAD; Constraint; Data exchange; Feature

1. Introduction

Computer support collaborative work (CSCW) has been applied in many manufacturing enterprises. This frequently results in the exchange of information among designers. As the most important media recording design ideas of designers, CAD models should flow among designers, whereas the differences of data structure between CAD systems decrease the accuracy of data exchange between them. To solve this problem, many researchers or organizations provide many different solutions. All the researches include the STEP-based method [1, 2], the Brep-based method [3] and the mesh-model-based method [4]. However, there is a common problem in these methodologies in that the target models obtained are not editable. Hoffman performed innovative research in this problem at Purdue University [5], and he proposed the feature-based data exchange method between heterogeneous CAD systems. The basic thinking of the feature-based method is to abstract features from a CAD model according to design history and rebuild them corresponding to the modeling history [6-11]. In this paper, feature-based data eXchange between Heterogeneous CAD Systems is abbreviated as FBDX. This method ensures that the target model is consistent with the source one in geometry and is editable too. However, some information about design intent is lost as the method is used to transform the CAD model. The so-called design intent is the relationship among functions of product, constraints, technique information and geometric information, and it represents the decision of designers [12]. Kim analyzed the loss problem about design intent in his paper [7], and considered that features, parameters,

† This paper was recommended for publication in revised form by Associate Editor Dae-Eun Kim
*Corresponding author. +86 411 8470 1894
E-mail address: tieqiang_ma@yahoo.com.cn

constraints and design history are the important impact factors for design intent.

At same time, Kim pointed out that constraints specify relationships between elements of a model that are required to be maintained if the model is edited [7]. So constraint is important to keep the geometric and topological stability of the CAD model, and constraints in the source model should be transformed into the target model during heterogeneous CAD data exchange.

2. Problem analysis

The constraint added to features or used to define the relationship between features is termed a feature constraint. A feature constraint includes geometric constraint, positioning constraint and topological constraint. Here, feature constraint is still called a constraint. The problems on the transformation of the feature constraint are as follows:

(1) There are several different transformation ways for a feature. The constraint of the target model cannot reflect the design intent, if an incorrect way is used to transform constraints.

(2) The geometric constraint of feature includes implicit geometric constraint and explicit geometric constraint, and the implicit one is added on features by the CAD system automatically [13]. The implicit constraint is often omitted during heterogeneous CAD data exchange, which results in the target model being under-constrained.

(3) The positioning constraint of a feature is often substituted by the coordinate of base point, which makes the feature fixed.

(4) A constraint acts on geometric elements. If the identifier of the geometric element is not exclusive, the reconstruction of feature constraint will encounter a mistake.

All the problems lead to the inconsistency of design intent between source model and the target one. Furthermore, current CAD systems handle the under-constrained target model in two different ways.

(1) An under-constrained model is permitted in some CAD systems. If a user edits the under-constrained model and changes its parameters, the model's shape may be changed in all feasible ways.

(2) In most CAD systems, an under-constrained model is not permitted. If the model is imported into this kind of CAD system, constraints will be added to it automatically and the model is made fully constrained.

Table 1. Meaning of symbols.

Symbol	Explanation	Symbol	Explanation
S_A	CAD system A	m	Number of features decomposed from F_A
S_B	CAD system B	FT	Function expressing the type of feature
M_A	Model built in S_A	FTS_A	Feature type set of S_A
M_B	Model built in S_B	FTS_B	Feature type set of S_B
F_A	Arbitrary feature in M_A	e_A	Arbitrary boundary geometric element of M_A
F_B	Arbitrary feature in M_B	e_B	Arbitrary boundary geometric element of M_B
$F_{B,k}$	The *k*th feature decomposed from F_A	[]	Optional

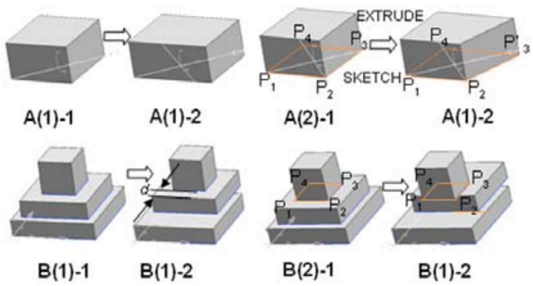


Fig. 1. Experiments of design intent loss.

The two ways cannot ensure that constraints in target model are the same to that in source model, which will make the design intent of the two models inconsistent. In case of shutting off the function of "Automatic Constraint" of UG, problems (2) and (3) are simulated by experiments, and the result of the experiments is shown as Fig.1. Figure 1-A shows the loss of implicit geometric constraint of the bottom, when the BLOCK feature is decomposed into the sketch feature and sketch-based extrude feature. While the BLOCK feature is transformed into the sketch feature and sketch-based extrude feature in figure 1-B, the positioning constraint of the corner point is substituted by its absolute coordinate, which results in the loss of positioning constraint. The two examples show that the constraint is often neglected in FBDX; thus the research on constraint transformation is significant in FBDX.

This paper proposes the feature constraint transformation method of FBDX in the following sections, and the effectiveness of the method will be tested by a data exchange experiment.

3. Constraint conversion

3.1 Symbol definition

Table 1 shows the meanings of symbols appearing in following sections.

3.2 Feature transformation

Aiming at transforming feature constraints, it is necessary to research feature transformation ways, which is the basis of feature constraint transformation. M_A is an arbitrary model built by S_A. If M_A is transferred to S_B as M_B, the transformation ways of arbitrary feature F_A in M_A are as follows.

As the modeling functions of S_A are the same as that of S_B in some degree, the intersection of feature type set of S_A and S_B $FTS_A \cap FTS_B \neq \Phi$ and two basic definitions were given according to the relations between FTS_A and FTS_B.

Definition 1 Basic Feature Type (BFT). Feature types in $FTS_A \cap FTS_B$ are defined as the basic feature types of S_A or S_B, and $FTS_A \cap FTS_B \subseteq$ *datum feature* $\cup$ *spline or surface* $\cup$ *sketch* $\cup$ *sketch-based feature* $\cup$ *simple design feature* $\cup$ *detail feature*.

Definition 2 eXtended Feature Type (XFT). Feature types in $FTS_A \oplus FTS_B$ are defined as the extended feature types of S_A or S_B, and $FTS_A \oplus FTS_B \subseteq$ *voxel feature* $\cup$ *complex design feature.*

Generally, the modeling functions of BFTs can be substituted by the BFTs'. The method of feature transformation was put forward according to the relations between BFTs and XFTs, and it will be discussed in two cases.

(1) BFT $\rightarrow$ BFT

If $FT\ (F_A) \in FTS_A \cap FTS_B$, then $FT\ (F_A) \in FTS_B$. If F_A is transferred to F_B ($F_B \in M_B$), then $FT\ (F_B)=FT\ (F_A)$ and $FT\ (F_B) \in FTS_B$. The transformation from F_A to F_B is called as 1:1 transformation, tagged as $F_A \xrightarrow{1:1} F_B$. Since BFTs occupy most of all feature types, 1:1 transformation should be the main transformation.

(2) XFT $\rightarrow$ BFTs

If $FT\ (F_A) \in (FTS_A - FTS_A \cap FTS_B)$, then $FT\ (F_A) \notin FTS_B$. Because $FTS_A - FTS_A \cap FTS_B \subseteq FTS_A \oplus FTS_B$, the modeling function of $FT\ (F_A)$ can be taken placed by BFTs in S_A. Thus F_A should be transformed as m features $F_{B,k}$ ($1 \le k \le m$), and $FT\ (F_{B,k}) \in FTS_A \cap FTS_B$. The transformation from F_A to $F_{B,k}$ ($1 \le k \le m$) is called a decomposing transformation, tagged as $F_A \xrightarrow{decomposed} \{F_{B,k} |\ FT\ (F_{B,k}) \in FTS_A \cap FTS_B\ (1 \le k \le m)\}$. The following is the decomposing transformation method presented by author.

The first decomposing transformation way can be described as the following set relationship.

$F_A \xrightarrow{decomposed} \{\ [F_{B,1}], F_{B,2}, F_{B,3},\ [F_{B,4}] | FT\ (F_{B,1}) \in$ *datum feature*, $FT\ (F_{B,2}) \in$ *sketch*, $FT\ (F_{B,3}) \in$ *sketch-based feature*, $FT\ (F_{B,4}) \in$ *detail feature*$\}$.

$F_{B,3}$ is a *sketch-based feature. The sketch-based feature* includes *extrude feature*, *revolve feature*, *sweep feature* and *blend feature*. If F_A can be transformed as different *sketch-based features*, the priorities will be *extrude feature*, *revolve feature*, *sweep feature* and *blend feature.*

The second decomposing transformation way can be described as follows.

$F_A \xrightarrow{decomposed} \{F_{B,1}, F_{B,2}, \ldots, F_{B,m} |\ (FT\ (F_{B,1})=FT\ (F_{B,2})=\ldots=FT\ (F_{B,m})) \wedge\ (FT\ (F_{B,k}) \in$ *simple design feature* $\vee\ FT\ (F_{B,k}) \in$ sketch-based feature $\vee\ FT\ (F_{B,k}) \in$ *spline or surface* $\vee\ FT\ (F_{B,k}) \in$ *detail feature*$) \wedge (1 \le k \le m))\}$.

Moreover, the CAD system provides special design features for designing an injection mold, progressive die, machine tool and so on.. Methodologies above will be combined, as these design features are transformed. The following is the transformation way of "T" slot, which is used to design a worktable of a machine tool.

T slot $\xrightarrow{decomposed}$ *{sketch,extrude,datum plane, sketch, extrude }*

To FBDX, application of methodologies mentioned above is the precondition that feature constraints are transformed correctly.

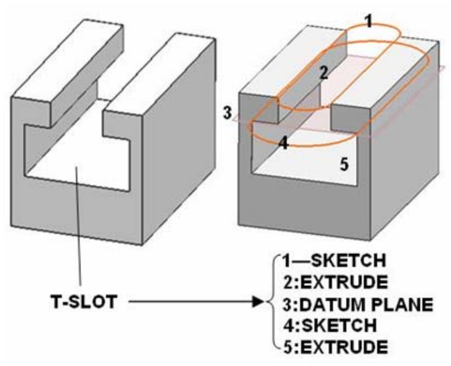


Fig. 2.Transformation method of *T slot.*

3.3 Identifier of geometric element

The geometric element is the acting object of the constraint, and its identification has been the focus of discussion. Researchers put forward many valuable solutions to this problem. In all solutions, the geometric certificate proposed by Hoffman is the most influential. The paper proposed an identification method of geometric element based on an improved geometry certificate. The geometry certificate of a geometric element was defined as a triple (*L, T, P)* [14]. *L* is the sequence number of geometric elements in list of points, lines or faces. *T* is the type of geometric element which includes points, lines and faces. *P* is the characteristic point of the geometric element. While the receiving CAD system is different from the sending one, the accuracy is bad and the efficiency is low if a geometry certificate is used to identify the geometric element. This paper improves the geometric element, and uses a quadruple (*F, L, T, P)* to identify the geometric element. *F* is the identifier of the feature, and the geometric element identified lies on the boundary of the feature. *T* is the type of geometric element. *L* is the sequence number of geometric elements in *T* list of *F*. *P* represents the characteristic parameters.

Fig. 3 shows an experiment about the stability of the feature's boundary. In Fig.3, the boundary face lists of BLOCK in both A and B are $\{f_0,f_1,f_2,f_3,f_4,f_5\}$ and $\{s_0,s_1,s_2,s_3,s_4,s_5\}$, respectively. It is shown that both the space positions of the faces in $\{f_0,f_1,f_2,f_3,f_4,f_5\}$ are the same as the space positions of the faces in $\{s_0,s_1,s_2,s_3,s_4,s_5\}$. More experiments have been done, and the results confirmed that the order of arrangement of geometric elements in the feature is stable. So *L* can be used to locate the geometric element in isomorphism CAD systems.

The so-called feature parameter is the parameter which can determine the space position and the geometric characteristic of geometric elements. For locating the geometric element with feature parameter, the feature parameter of geometric element is defined as $\{x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}, p_1, p_2, \ldots, p_n\}$. $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}$ and z_{max} represent the boundary box, and n is the number of geometric parameters. For different kinds of geometric elements, their geometric parameters are also different. For example, the geometric parameters of line are the end points, and those of arc are the end points and the center point. Fig.4 shows the feature parameters of arc.

As using this kind of geometry certificate to identify geometric elements, the feature including the geometric element is found first by *F*. And then *L*, *T* and *P* are used to further find the correct geometric element. The usage of (*F, L, T, P)* limits directly the range of geometric element to the feature in which the geometric element lies. So the positioning efficiency of the geometric element will be higher when using (*F, L, T, P)* to identify the geometric element. Furthermore, the positioning accuracy of the geometric element will be promoted when the characteristic point is substituted by characteristic parameters. The identification method will be applied in feature constraint conversion in FBDX.

3.4 Constraint conversion

In this section, the method of feature constraint conversion is given through analyzing the transformation method of different kinds of features. Since the 1:1 transformation is simple, this paper will only discuss the feature constraint conversion method of implicated feature. In order to ensure that the design intent expressed by $F_{B,k}$ $(1 \le k \le m)$ is the same as that of F_A, the principles about feature constraint conversion are presented.

(1) There must be equivalent constraints to the feature constraints of F_A in $F_{B,k}$ $(1 \le k \le m)$.

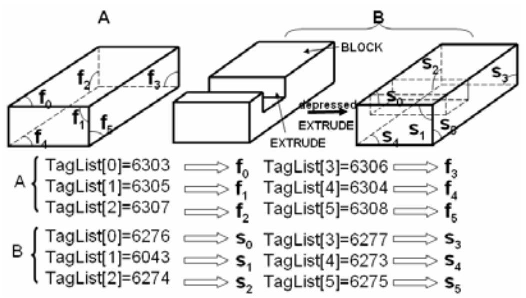


Fig. 3. Experiment about the stability of feature's boundary.

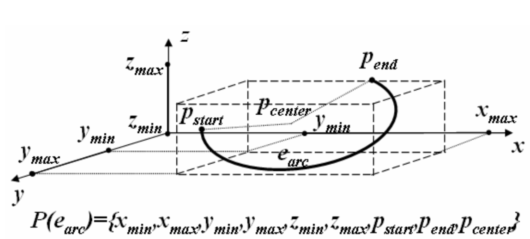


Fig. 4. Feature parameters of arc.

(2) while $F_{B,k}$ $(1 \le k \le m)$ is edited, $F_{B,k+1}$, $F_{B,k+2}$,…, $F_{B,m}$ must change with $F_{B,k}$ $(1 \le k \le m)$.

(3) $F_{B,k}$ $(1 \le k \le m)$ must be constrained completely.

The conversion methodologies of geometric constraint, positioning constraint and topological constraint are provided, according to these principles. The conversion method of positioning constraint and that of topological constraint will be discussed together because of the mutual transformation relationship between positioning constraint and topological constraint transformed.

(1) The conversion method of geometric constraint

Geometric constraint is used to constrain the position relationship between geometry elements of a sketch. The sections of implicated features are divided into five primitive sections in Fig. 5.

Implicated features will be decomposed into primitive-section-sketch-based features or their combination. For pocket feature with fillets, its section cannot be expressed by primitive section directly. So it is necessary to get rid of the fillets from the pocket, and transform the pocket into a combination of primitive-section-sketch-based features and fillets. While rebuilding a primitive-section-based sketch, the following geometric constraints should be added to sketches.

(a) perpendicular $(e_1, e_2) \cap$ *perpendicular* $(e_2, e_3) \cap$ *perpendicular* (e_3, e_4)

(b) equal $(e_2, e_3) \cap$ *angle* (e_2, e_3)

(c) equal $(e_2, e_4) \cap$ *angle* $(e_2, e_3) \cap$ *angle* (e_3, e_4)

(d) No geometric constraint

(e) tangent $(e_1, e_2) \cap$ *tangent* $(e_2, e_3) \cap$ *tangent* $(e_3, e_4) \cap$ *parallel* (e_1, e_3)

In (a)-(e), *perpendicular ( ), equal ( ), angle ( ), parallel ( ) and tangent ( )* represent the constrain functions among geometric elements *e*.

(2) Conversion method of positioning and topological constraint.

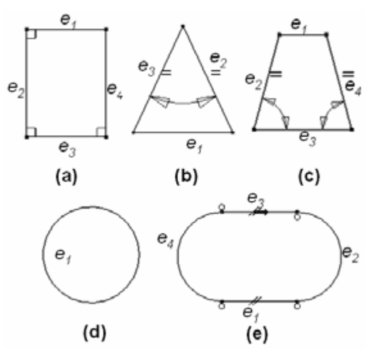


Fig. 5. Primitive sections of implicated feature.

First, if the section of F_A is parallel to the link face of F_A, the sketch feature transformed from F_A will attach on the face corresponding to the link face of F_A. The transformation way of F_A is as follows.

$F_A \xrightarrow{decomposed} \{F_{B,1}, F_{B,2}, \ldots | FT\ (F_{B,1}) \in$ *sketch feature*, $FT\ (F_{B,2}) \in$ *sketch-based feature*, $\ldots\}$.

Sketch feature $F_{B,1}$ succeeds the topology of constraint of F_A, and sketch attaches on the face corresponding to the link face of F_A. The level and vertical positioning constraints of $F_{B,1}$ are transformed from the positioning constraints of F_A; the reference geometric elements of positioning constraint are corresponding to the reference elements on which the positioning constraints of F_A attach.

Secondly, if the section of F_A is perpendicular to the link face of F_A, the sketch feature transformed from F_A must attach on a datum plane. The transformation way of F_A is as follows.

$F_A \xrightarrow{decomposed} \{F_{B,1}, F_{B,2}, F_{B,3}, \ldots | FT\ (F_{B,1}) \in$ *datum plane*, $FT\ (F_{B,2}) \in$ *sketch feature*, $FT\ (F_{B,3}) \in$ *sketch-based feature*, $\ldots\}$.

One of the positioning constraints of F_A will be transformed as the positioning constraint of the datum plane, and the other will be transformed to position the *sketch feature* in one direction. The topological constraint of F_A will be converted as the positioning constraint in the other direction which is vertical to the former.

Some of the constraint conversion methodologies are used in the example in Fig. 6, which shows that the design intents are kept well.

3.5 Global constraint conversion algorithm

This section gives the algorithm of global constraint conversion. In order to describe the method, some definitions are given. If *ID* (e_A) is defined as the identification function *e*, then the following is the definition of identifier mapping set.

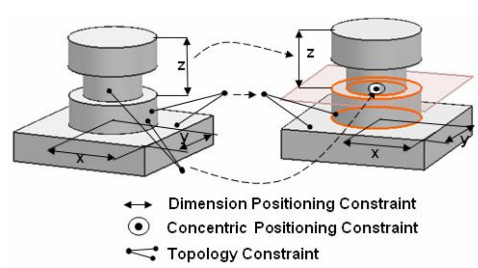
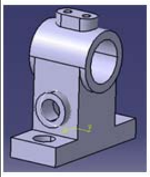
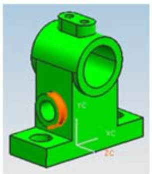
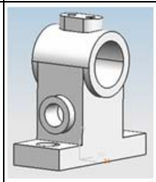
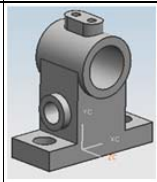
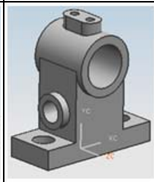
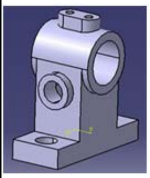
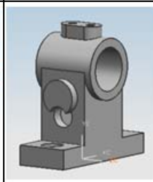
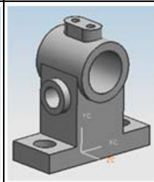
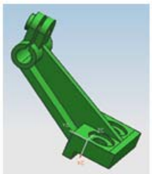
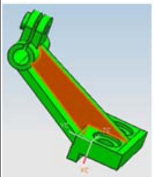
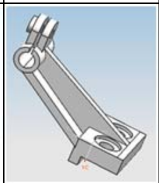
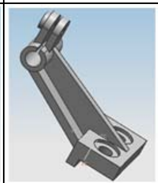
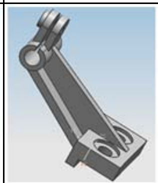
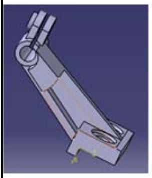
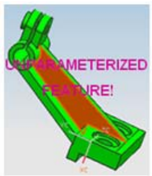
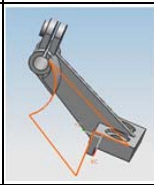


Fig. 6. Example of feature constraint conversion.

Table 2. Results of data exchange experiment.

Source model			Data exchange experiment				
Part	Status	Catia	Step AP203	Iges	*.Model	Reference [11]	This paper
1	initial model						
	Variant model						
2	Initial model						
	Variant model						

0 Begin;
1 Load M_A into CAD System S_A;
2 Travel the feature tree of M_A, and the list of features is accessed;
3 Get the amount of features, and store it in variant N;
4 Define variant i and j , and $i=0, j=0$;
5 Do while $i \leq N$
It is assumed that arbitrary feature F_A is the i^{th} feature of M_A;
$j=i$;
Do while $j \leq N$
It is assumed that arbitrary feature F_A' is the j^{th} feature of M_A;
If constraints of F_A' reference on e_A which is the boundary geometric elements of F_A, then
If the constraint is a positioning constraint or a topology constraint, then
Depress F_A' and all other features after F_A'
Else if the constraint is a geometric constraint
Depress all features after F_A;
End if
Abstract the identifier of e_A, $ID(e_A)$;
In M_B, e_B is the corresponding geometric element of e_A, *and* $ID(e_B) = f(ID(e_A))$
Push(<$ID(e_A)$, $ID(e_B)$>,&*IMS*);
End if
j=j+1;
End While
It is assumed that arbitrary feature F_B is the corresponding feature of F_A in M_B
If the constraints of reference on geometric element e_B, and the identifier of e_A is given as $ID(e_A)$, then
Find($ID(e_A)$,&$ID(e_B)$);
End if
Add constraints to F_B.
i=i+1;
6 End while
7 End;

Fig. 7. Global constraint conversion algorithm.

Definition 3 Identifier Mapping Set. For an arbitrary geometric element $e_A \in M_A$, e_B is its corresponding geometric element in M_B. There must be an injection f from M_A to M_B, so that $ID\ (e_B)=f\ (ID\ (e_A))$. The mapping relationship from $ID\ (e_A)$ to $ID\ (e_B)$ is represented as ordered couples $<ID\ (e_A), ID\ (e_B)>$ called identifier mapping about e_A, and all the identifier mappings form the Identifier Mapping Set in the transformation from M_A to M_B. Identifier Mapping Set is simply denoted as *IMS*.

Two functions are defined on *IMS*.

(1) *Push* ($<ID\ (e_A), ID\ (e_B)>$,&*IMS*). The function is used to add $<ID\ (e_A), ID\ (e_B)>$ into *IMS*, and return the refreshed *IMS* simultaneously.

(2) *Find* ($ID\ (e_A)$, &$ID\ (e_B)$). The function is used to search for $<ID\ (e_A), ID\ (e_B)>$ by $ID\ (e_A)$ in *IMS*, and return $ID\ (e_B)$.

The algorithm of global constraint conversion is as shown in Fig. 7.

The constraint conversion from M_A to M_B will be realized by the algorithm. The validity of this method will be verified by further experiments.

4. Experiments

A data exchange interface from CATIA to UG has been developed by VC++6.0, and the method of constraint conversion was applied in the interface. The interface will be compared with other four different interfaces to verify the method proposed. The four interfaces are STEP AP203, IGES, *.MODEL and the interface developed according to reference [11], respectively. Table 2 shows the experimental results of the contrast experiment.

For every part in Table 2, models in the first row are the initial model and those in the second row are the parametric variant model. For part models transformed from different data exchange interfaces, the first part model expresses the difference of the positioning constraint, and the second part model represents the difference of the geometric constraint.

Conclusions are drawn by analyzing results of the five experiments.

(1) Uneditable target models are obtained by interfaces for STEP, IGES and *.MODEL, and design intent is seriously lost in the process of CAD data exchange.

(2) If some constraints are ignored in FBDX, parametric variant characteristics of the target model will be different from that of the source one. This will result in the loss of design intent in heterogeneous CAD data exchange.

(3) If the data exchange interface is developed in the method mentioned by this paper, the parametric variant characteristics of target model are same as that of the source one in CAD data exchange. This shows that consistency of design intent is maintained well and the constraint conversion method proposed is effective.

5. Conclusion

This paper studied the loss of design intent in FBDX, and put forward a constraint conversion method, which is proven to be an effective solution to this problem by data exchange experiments from CATIA to UG. Its advantages are as follows.

(1) The parametric variant characteristics of the model will be maintained well, if the data exchange interface with the constraint conversion method mentioned in this paper is used to transform CAD model.

(2) The constraint conversion method mentioned in this paper can be applied to develop the data exchange or transmission tools for collaborative design users, and is valuable for improving the interoperability between users working on heterogeneous CAD systems.

(3) The method is a supplement to the theory of feature-based data exchange between heterogeneous CAD systems.

Acknowledgments

This research is sponsored by the Program for New Century Excellent Talents in University, Ministry of Education, China (Grant No. NCET-05-0285).

References

[1] X. A. Chen and H. Tan, Research on data exchange method of neutral file for 3D geometric model. *China J. of Mech. Eng.*, 37 (10) (2001) 93-95, 99.

[2] H. J. Yang and R. X. Ning, Method on 3D model data exchange between Pro/E and UG. *J. Comput. Aided Design & Comput. Graphics*, 17 (12) (2005) 2756-2759.

[3] Q. F. Wang, D.Gao, Y. Huang et al. On the data transfer interface for 3D CAD system. *Mech. Sci.*

and Tech. for Aero. Eng., 24 (12) (2005) 1478-1481.
[4] Y. H. Liu, J. Liu and L. P. Chen, Lightweight representation for 3D product model. *J. Comput. Aided Design & Comput. Graphics*, 18 (4) (2006) 602-607.
[5] C. M. Hoffman and R. Juan, Erep-An Editable High-Level Representation for Geometric Design. *Geometric Modeling for Production Realization*, P. Wilson, M. Wozny, M. Pratt (eds), North Holland, 1993:129-164.
[6] Y. H. Liu, L. P. Chen and F. Y. Zhong, Exchange feature information of heterogeneous CAD systems by design history, *China Mech. Eng.*, 14 (21) (2003) 1845-1847.
[7] Junhwan Kim, Michael J. Pratt, Raj G. Iyer et al. Standardized data exchange of CAD models with design intent. *Computer-Aided-Design*, (2007) http://dx.doi.org/10.1016/j.cad.2007.06.014.
[8] Michael J. Pratt, Bill D. Anderson and Tony Ranger. Towards the standardized exchange of parameterized feature-based CAD models. *Computer-Aided Design*, 37 (12) (2005) 1251-1265.
[9] Duhwan Mun, Soonhung Han, Junhwan Kim and Youchon Oh. A set of standard modeling commands for the history-based parametric approach. *Computer-Aided Design*, 35 (13) (2003) 1171-1179.
[10] G. Choi, D. Mun and S. Han, Exchange of CAD part models based on the macro-parametric approach. *Intl J. CAD/CAM*, 2 (2) (2002) 23-31.
[11] X. D. Shao, F. Chen and C. G. Xu, Study on isomerous CAD model exchange based on feature. *China Mech. Eng.*, 18 (1) (2007) 60-64
[12] H. Jiang, T. Wang, Q. Cao and G. L. Xiong, Methodology of design history modeling. *Mech. Sci. Tech. for Aero. Eng.*, 19 (2) (2000) 197-200, 209.
[13] J. J. Shah, M. Mäntylä, Parametric and feature based CAD/CAM: Concepts, techniques and applications, *John Wiley & Sons New York, USA,* (1995).
[14] C. M. Hoffman, R. Joan-Arinyo, CAD and the product master model, *Computer-Aided Design*, 30 (11) (1998) 905-918.

Sun Wei is a professor and the deputy dean of school of mechanical engineering at Dalian University of Technology. His research interests include knowledge-based product digital design, computer supported collaborative design, analysis and optimization of product. He received a BS, a MS and a Ph.D in mechanical manufacturing and automation from Dalian University of Technology, P. R. China.

Ma Tie-Qiang is a Ph.D. candidate in school of mechanical engineering at the Dalian University of Technology. He received a BS and a MS from Dalian JiaoTong University P. R. China. His research interests include computer graphics &computer aided design, product data management, computer supported collaborative design.

Huang Yu-Jun is a graduate in School of mechanical engineering at the Dalian University of Technology. He received a BS from Wuhan University of Science and Technology P.R. China. His research interests include product data exchange and computer graphics.