6

Representing Dimensions and Features in a Product Model

Tsuzuki, M. S. G.; Miyagi, P. E.; Moscato, L. A.
Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia Mecânica/Mecatrônica
Av. Prof. Mello Moraes, 2231
CEP 05508-900 - São Paulo - SP - BRAZIL
Tel: +55-11-818-5565
Fax: +55-11-813-1886
email: mtsuzuki@cat.cce.usp.br

Abstract

This work is based on two believes about the next generation of Computer Aided Design: first, it is necessary to *explicitly store features and dimensions* in a Product Model; second, it is necessary to *support user defined features*. The Product Model is represented as a hierarchical structure where it is possible to define two kinds of dimensions: *local dimensions* and *relative dimensions*. Relative dimensions are constraints that associate two different nodes in the hierarchical structure. It is also proposed an algorithm to satisfy a set of relative dimensions.

Keywords

Product Model, Form Feature, Parametric Design, Dimension Representation

1 INTRODUCTION

The state of research of Computer Aided Design can be identified as a transition from design based on geometry to design based on features. In spite of several works related to feature representation, a common sense has not been found. Several works have discussed the usability of the CSG representation and/or the B-Rep representation. The B-Rep representation is preferred because vertices, edges and faces are explicitly represented (Wilson, 1990). However, the CSG modeling approach is most suitable for dimension driven geometry approach (Gossard, 1988).

We are working to create a Design by Feature System in which an user can generate a design by attaching features to the product. This system is based on two believes about the next generation of Computer Aided Design: first, it is necessary to **explicitly store features and dimensions in a Product Model**; second, it is necessary to **support user defined features**.

Q. Sun et al. (eds.), *Computer Applications in Production Engineering*

Ovtcharova (1992) proposed that a Design by Feature Systems has four levels of abstraction: application level, feature definition level, feature representation level and geometric modeller level. This work is part of the implementation of the third level: representation of features.

2 PRODUCT MODEL

The Product Model is represented through a hierarchical tree structure with several levels of complexity (see Figure 1). The Product Model Node represents the root node. Every node has access to all information related to the nodes immediately below. In this way, the Assembly has access to all information related to the Parts immediately below it. In this structure there is a difference between the leftmost node and the others. All information related to the leftmost node is transferred to the upper node directly. All information related to the other nodes is transferred to the upper node based on transformations. Because of this property, we say that the leftmost node is fixed and the other nodes are moveable. Every node has a set of primitive constituent elements that can be a set of **points**, **lines** or **planes**.

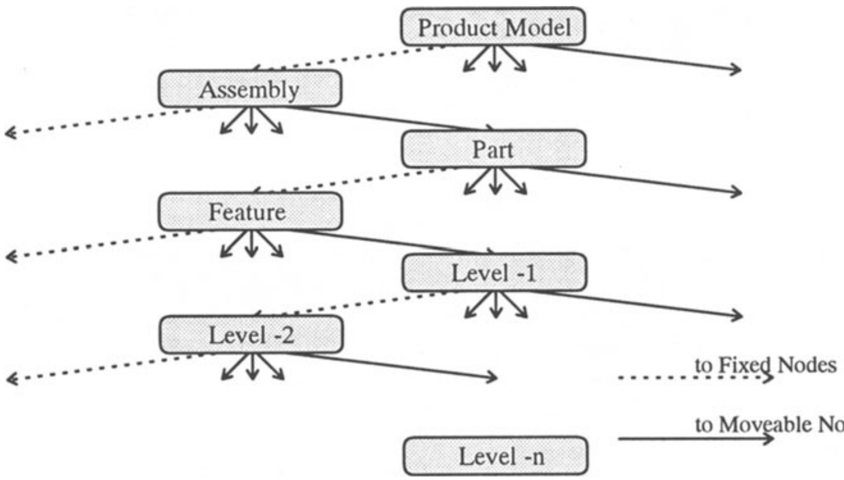

Figure 1 The hierarchical representation of the Product Model.

According to this structure, it is possible to define two types of dimensions: **local dimensions** and **relative dimensions**. Dimensions associated to one unique node are local dimensions. A relative dimension constrains primitive constituent elements from two different nodes, one immediately below and another immediately above. The fixed node has no relative dimension constraining it to the immediately above node as its information is transferred without any transformation.

A node is symbolically represented by:

$$A = F(B_1, t_2 \cdot B_2, t_3 \cdot B_3, \ldots, t_n \cdot B_n) \quad (1)$$

where A is the upper node, B_i is an inferior node of node A and t_i is a transformation applied to node B_i. B_1 is the leftmost node. Only node B_1 does not have a transformation associated with. Function F represents the transferring of information from the inferior node to the upper node. This work will define an algorithm that will find all the transformations t_i such that the nodes are correctly positioned, i.e., all the relative dimensions are satisfied.

3 REPRESENTING FEATURES

Our objective is to define a mechanism to facilitate the manipulation of features by the user. Such that the user selects a feature from the menu and just provides some primitive constituent elements to attach a feature in the Product Model. This effect can be implement by **associating some relative dimensions to features**. Figure 2 illustrates an example of feature definition.

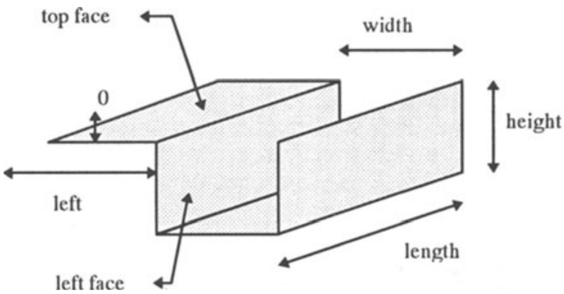

Figure 2 Example of feature definition.

Where width, height and length are feature parameters, and left and 0 are relative dimensions. The relative dimension 0 represents that the top face must have distance 0 (zero) to a Part's face when the feature is attached to the Part. The value of the relative dimension left must be supplied when the feature is attached to a part. The relative dimension left represents that the left face must be at some distance from a Part's face.

4 SPECIAL SETS OF TRANSFORMATIONS

4.1 Relative Dimensions

A set of relative is represented by:

$$\Delta = \{ d_1, d_2, \cdots, d_n \} \quad (2)$$

where, Δ is a set of relative dimensions and d_i is a relative dimension. A relative dimension can be in two different states: satisfied and not satisfied. The inferior node is positioned relatively to the upper node such that all the relative dimensions are satisfied. We will represent the state of the set of relative dimensions by:

$$A <\Delta> B = \delta \quad (3)$$

where, A is the upper node, B is an inferior node of node A and δ is the set of satisfied relative dimensions. In this way, if $\delta = \{\ \}$ then no relative dimension is satisfied. If $\delta = \Delta$ then all relative dimensions is satisfied. All the fifteen types of relative dimensions are listed in Table 1.

Table 1 All the fifteen types of relative dimensions

relative dimensions type	code
plane-plane distance, with coincident normals	DisCN
plane-plane distance, with opposite normals	DisON
plane-plane angle	AngPP
plane-line distance (and vice versa)	DisPR (DisRP)
plane-line angle (and vice versa)	AngPR (AngRP)
plane-point distance (and vice versa)	DisPp (DispP)
line-line distance, with coincident directions	DisCD
line-line distance, with opposite directions	DisOD
line-line angle	AngRR
line-point distance (and vice versa)	DisRp (DispR)
point-point distance	Dispp

4.2 Set of Transformations

The transformation of a node can be represented by:

$$E' = t \cdot E \tag{4}$$

where, t is the transformation, E represents a node and E' represents the transformed node. Now, it is possible to define two operations over sets of transformations: **intersection** and **distribution**. The intersection operation is defined as:

$$t \in \varphi_1 \text{ and } t \in \varphi_2 \Rightarrow t \in \varphi_1 \cap \varphi_2 \tag{5}$$

where, φ_1 and φ_2 represent two sets of transformations, t is a transformation and $\cap$ is the intersection operator. The distribution operation is defined as:

$$t_1 \in \varphi_1 \text{ and } t_2 \in \varphi_2 \Rightarrow t_1 \cdot t_2 \in \varphi_1 * \varphi_2 \tag{6}$$

where, φ_1 and φ_2 represent two sets of transformations, t_1 and t_2 are transformations and $*$ is the distribution operator. Note that this operation is not commutative.

4.3 Arbitrary Coordinate

Two nodes A and B are **equivalent**, if every internal point of A is internal of B and if every external point of A is external of B and if every point in the boundary of A is in the boundary of B. Given a transformation t_{ST} such that:

$$E' = t_{ST} \cdot E. \tag{7}$$

Then, if the node E is equivalent to node E' and the transformation t_{ST} is not the identity, then it is said that E has a set of arbitrary coordinates. The set of all transformations t_{ST} is called set of symmetric transformations. This can be better understood through an example: a cylinder can be rotated through its axis of symmetry that the result of this transformation is equivalent to the original cylinder. Particularly, the primitive constituent elements plane, line and point have sets of symmetric transformations that will play an important role in our algorithm:

- **plane**: it is possible to rotate a plane through any axis parallel to its normal vector or to translate it in any direction normal to its normal vector that the result of the transformation is equivalent to the original plane. The plane's set of symmetric transformations is represented by *PLN* ;
- **line**: it is possible to rotate a line through itself or to translate it according to its direction that the result of the transformation is equivalent to the original line. The line's set of symmetric transformations is represented by *RET*;
- **point**: it is possible to rotate a point through any axis that pass through itself that the result of the transformation is equivalent to the original point. The point's set of symmetric transformations is represented by *PNT* .

4.4 Degrees of Freedom Associated with Relative Dimensions

A relative dimension has degrees of freedom in the sense that the inferior node can be transformed and the satisfied relative dimensions will remain satisfied. Consider that a transformation $t^{\{d1\}}$ [1] satisfies the relative dimension d_1. Symbolically represented by:

$$A < \Delta > (t^{\{d1\}} \cdot B) = \{ d_1 \}. \tag{8}$$

The set of all transformations t_{DF} that maintain the relative dimension d_1 satisfied is called set of degrees of freedom associated to the relative dimension d_1. Symbolically represented by:

$$A < \Delta > (t_{DF} \cdot t^{\{d1\}} \cdot B) = \{ d_1 \}. \tag{9}$$

The Greek letter τ will be used to represent the sets of degrees of freedom. The sets of degrees of freedom are associated to the set of symmetric transformations of the primitive constituent elements constrained by the relative dimensions. As a relative dimension d_i constrains two primitive constituent elements, one is associated to the upper node and the other is associated to the inferior node. The sets of symmetric transformations are going to be represented by τ_S^{di} and τ_I^{di}, where the first is associated to the upper node and the second is associated to the inferior node. The set of degrees of freedom associated to the relative dimension d_i is represented by:

$$DF(d_i) = \{ \tau_S^{di}, \tau_I^{di} \}. \tag{10}$$

In this way, a relative dimension has two subsets of degrees of freedom. If

$$A < \Delta > B = \{ d_i \} \tag{11}$$

then transforming the inferior node B by any $t_I \in \tau_I^{di}$, the relative dimension d_i will remain satisfied. Symbolically:

[1] The index represent that the relative dimension d_1 is satisfied by this transformation.

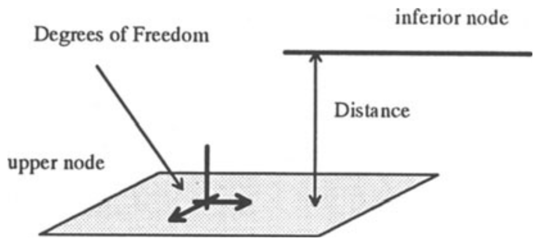

Figure 3 A relative Dimension of type plane-line distance.

Table 2 Sets of Degrees of Freedom associated with the Relative Dimensions

Relative Dimension	Set of Degrees of Freedom
DisNC	*{ PLN , PLN }*
DisNO	*{ PLN , PLN }*
AngPP	*{ PLN * 3DT , PLN *3DT }*
DisRP	*{ RET , PLN }*
DisPR	*{ PLN , RET }*
AngRP	*{ RET *3DT , PLN *3DT }*
AngPR	*{ PLN *3DT , RET *3DT }*
DisPp	*{ PLN , PNT }*
DispP	*{ PNT , PLN }*
DisOC	*{ RET , RET }*
DisOO	*{ RET , RET }*
AngRR	*{ RET *3DT , RET *3DT }*
DispR	*{ PNT , RET }*
DisRp	*{ RET , PNT }*
Dispp	*{ PNT , PNT }*

$$A < \Delta > (t_I \cdot B) = \{ d_i \}. \tag{12}$$

If the upper node is transformed by any $t_S \in \tau_S^{di}$ then the relative dimension d_i will remain satisfied too. However, it is important to note that the relative positioning between the upper node and the inferior node is not altered if the upper node is transformed by t_S or if the inferior node is transformed by t_S^{-1} (inverse transformation). In this way, it is possible to represent the transformation of the upper node by the following expression:

$$A < \Delta > (t_S^{-1} \cdot t_I \cdot B) = \{ d_i \}. \tag{13}$$

This property creates a **hierarchy** between the subsets of degrees of freedom. When, the transformation t_S^{-1} is applied to the inferior node, then the inferior node's subset of symmetric transformations is modified by this transformation. Suppose that a relative dimension of type plane-line distance was satisfied (see Figure 3). In this case, it is possible to transform the line according to its symmetric transformations that the relative dimension will remain satisfied, and it is possible to transform the line according to the plane's symmetric transformations that the relative dimension will remain satisfied. The definition of the set of degrees of freedom associated to a relative dimension depends on its type and the type of the primitive constituent elements that it constrains.

Before listing all sets of degrees of freedom associated with relative dimensions it is necessary to define a special set of transformations that represent all the possible translations in the three dimensional space. This set of transformations is represented by *3DT*. Table 2 defines all the sets of degrees of freedom associated with the relative dimensions.

4.5 Degrees of Symmetry

Both upper and inferior nodes have a set of symmetric transformations. This set will be called set of degrees of symmetry. It is represented by:

$$X = \{ \sigma_S, \sigma_I \} \tag{14}$$

where, σ_S is the set of degrees of symmetry associated to the upper node and σ_I is the set of degrees of symmetry associated with the inferior node. To satisfy the *i*th relative dimension, it is necessary to determine the set of degrees of freedom $\varphi_{i+1} = \{ \tau_S , \tau_I \}$. The set of degrees of symmetry must be updated every step, such that at the *i*th step the set of degrees of symmetry is represented by $X_{i+1} = \{ \sigma_S , \sigma_I' \}$. Now, if the expression

$$\tau_S * \tau_I \subset \sigma_S * \sigma_I' \tag{15}$$

is true then the inferior node is fixed to the upper node.

5 SATISFYING A RELATIVE DIMENSION

We are interested in finding a transformation t^{Δ} such that:

$$A < \Delta > (t^{\Delta} \cdot B) = \Delta. \tag{16}$$

The inferior node B is relatively positioned by the transformation t^{Δ} to the upper node A such that all the relative dimensions are satisfied. In this work, we will define an algorithm that will find all these transformations that satisfy a set of relative dimensions. The algorithm will search for the transformations step by step, satisfying one relative dimension each time. In other words, we will find firstly a transformation $t^{\{d1\}}$[2]; secondly a transformation $t^{\{d2\}}{}_{\{d1\}}$[3] and continuing.

The relative dimensions are satisfied step by step. At the *i*th step a relative dimension d_i will be satisfied. The relative dimensions $d_1, d_2, \cdots, d_{i-1}$ are satisfied already. Associated to the *i*th step there is a set of degrees of freedom φ_i that maintain all the satisfied relative dimensions satisfied. Symbolically represented by:

---

[2]The index represent that the relative dimension d_1 is satisfied by this transformation.
[3]The indexes represent that the relative dimension d_2 is satisfied by this transformation and the relative dimension d_1 remains satisfied.

$$A < \Delta > B' = \{ d_1, d_2, \cdots, d_{i-1} \} \tag{17}$$

and,

$$\forall\, t \in \varphi_i \Rightarrow A < \Delta > (t \cdot B') = \{ d_1, d_2, \cdots, d_{i-1} \}. \tag{18}$$

Suppose that the relative dimension d_i is satisfied by a transformation:

$$T^{\{di\}}_{\{di-1, \ldots, d1\}} = \{ t^{\{di\}}_{\{di-1, \ldots, d1\},S}, t^{\{di\}}_{\{di-1, \ldots, d1\},I} \} \in \varphi_i \tag{19}$$

where the meaning of the subindexes $_S$ and $_I$ are, respectively, superior and inferior. In this way,

$$A < \Delta > (t^{\{di\}}_{\{di-1, \ldots, d1\},S} \cdot t^{\{di\}}_{\{di-1, \ldots, d1\},I} \cdot B') = \{d_1, d_2, \cdots, d_{i-1}, d_i \}. \tag{20}$$

The set of degrees of freedom associated with the inferior node must be actualized by the transformation $t^{\{di\}}_{\{di-1, \ldots, d1\},S}$. Then,

$$\varphi_i' = \{ \tau_S, t^{\{di\}}_{\{di-1, \ldots, d1\},S} \cdot \tau_I \} \tag{21}$$

The set of degrees of freedom φ_{i+1} is the intersection of φ_i' and $DF(d_i)$[4]. Symbolically:

$$\varphi_{i+1} = \varphi_i' \cap DF(d_i). \tag{22}$$

At this moment it is necessary to verify whether the expression 15 is true or not. If it is true then the algorithm is finished. If there is any remaining relative dimension then the nodes were overconstrained. According to equation 22 the intersection between sets of degrees of freedom plays an important role in the algorithm. All the possible cases of intersection are represented in table 4.

Table 4 Intersection between sets of symmetric transformations

	TOT	*PLN*	*RET*	*PNT*	*ROT*	*TRN*	*R3D*	*T3D*	*T2D*
TOT	*TOT*	*PLN*	*RET*	*PNT*	*ROT*	*TRN*	*R3D*	*T3D*	*T2D*
PLN		*PLN* *TRN*	*ROT* *TRN*	*ROT*	*ROT* *EMP*	*TRN* *EMP*	*PLN* *T2D*	*T2D*	*T2D* *TRN*
RET			*RET* *TRN* *EMP*	*ROT* *EMP*	*ROT* *EMP*	*TRN* *EMP*	*RET* *TRN*	*TRN*	*TRN* *EMP*
PNT				*PNT* *ROT*	*ROT* *EMP*	*EMP*	*ROT*	*EMP*	*EMP*
ROT					*ROT* *EMP*	*EMP*	*ROT* *EMP*	*EMP*	*ROT*
TRN						*TRN* *EMP*	*TRN*	*TRN*	*TRN* *EMP*
R3D							*R3D* *T3D*	*T3D*	*T2D*
T3D								*T3D*	*T2D*
T2D									*T2D* *TRN*

[4] $DF(d_i)$ is the set of degrees of freedom associated with the relative dimension d_i.

where *TOT*, *ROT*, *TRN*, *R3D*, *T2D* and *EMP* are sets of degrees of freedom. Their meaning are explained below:

- *TOT*, totally free;
- *PLN*, plane's set of symmetric transformations;
- *RET*, line's set of symmetric transformations;
- *PNT*, point's set of symmetric transformations;
- *ROT*, it is possible to rotate the node through an axis of rotation that the result of the transformation will be equivalent to the original;
- *TRN*, it is possible to translate the node through a direction that the result of the transformation will be equivalent to the original;
- *R3D*, it is possible to rotate the node through an axis of rotation and to translate it in any direction that the result of the transformation will be equivalent to the original;
- *T2D*, it is possible to translate the node through any direction perpendicular to an specific direction that the result will be equivalent to the original;
- *EMP*, totally fixed.

5.1 Solution Tree

To determine the transformations that satisfy a set of relative dimensions, we will define a hierarchical tree structure. In this tree there will be two kinds of level (see Figure 4). Nodes associated with odd levels represent relative dimensions to be satisfied, and nodes associated with even levels represent transformations that satisfy the relative dimensions associated to the upper node.

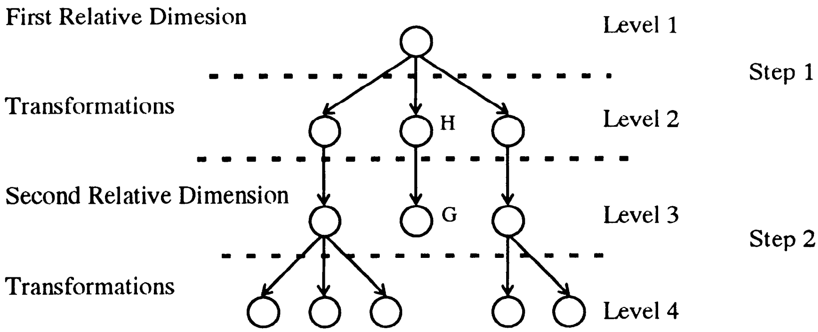

Figure 4 Solution Tree (the relative dimension associated with node *G* could not be satisfied).

The *i*th relative dimension to be satisfied is represented in level $2 \cdot (i-1)+1$. The transformations that satisfy a relative dimension d_i are represented in level $2 \cdot i$. This structure will allow to determine all the transformations that satisfy a given set of relative dimensions.

Figure 4 shows an example with a set of two relative dimensions where the algorithm found five transformations that can satisfy the set. It is not guaranteed that all the transformations are different. In this way, it is necessary to compare them and select only the different transformations. It is possible to observe that the relative dimension associated with node G could not be satisfied, so the transformation associated with node H is discarded.

6 FUTURE WORKS

We did not show, however it is possible to represent user defined features and to explicitly represent features and dimensions using the concept of relative dimensions (Tsuzuki, 1995). Another implication of the use of relative dimensions is that it is possible to *rationalize* the definition of dimensions in a project, as a consequence of the *encapsulation* of the information in the nodes of the structure shown in Figure 1. It is necessary to define a new representation where points, lines and planes are explicitly represented. We are working in the definition of some heuristics for satisfying relative dimensions. The method is efficient because only the branches in the tree that are affected by a given change need to be reevaluated. Underconstrained nodes can be identified by the method and they can be partially evaluated.

7 REFERENCES

Gossard, D.C.; Zuffante, R.P. and Sakurai, H (1988) Representing Dimensions, Tolerances and Features in MCAE Systems, *IEEE Computer Graphics & Applications*, March, 51-59.

Ovtcharova, J.; Pahl, G. and Rix, J. (1992); A Proposal for Feature Classification in Feature-Based Design, *Computer & Graphics*, 16(2):187-195.

Tsuzuki, M.S.G. (1995), A contribution to parametric feature representation, *Doctor Thesis (in Portuguese)*, Escola Politécnica da USP.

Wilson, P.R. (1990); Feature Modeling Overview, *Proceedings of the 17th International Conference on Computer Graphics and Interactive Techniques SIGGRAPH'90*, Course Notes 12, Dallas, XI.1-XI.56.

TSUZUKI, M. S. G., born in 1963. Graduated in Electrical Eng. from Escola Politécnica da USP (EPUSP) - Brazil in 1985. Received the M.Sc. from the Dep. of Information Processing Eng. from Yokohama National University - Japan in 1988. Assistant of the Dep. of Mechanics/Mechatronics Eng. of EPUSP since 1990.

MIYAGI, P. E. Ph.D., born in 1959, graduated in Electrical Eng. from EPUSP - Brazil in 1981. Received the M.Sc. and Ph.D. degrees from Tokyo Institute of Technology - Japan in 1985 and 1988 respectively. Associate Professor of Dep. of Mechanics/Mechatronics Eng. of EPUSP since 1993.

MOSCATO, L. A. Ph.D., born in 1946, graduated in Electrical Eng. from EPUSP - Brazil in 1969. M.Sc. and Ph.D. from Electrical Eng. Dep. from EPUSP in 1971 and 1978 respectively. Full Professor of Dep. of Mechanics/Mechatronics Eng. of EPUSP since 1988.