Int J Adv Manuf Technol (2000) 16:733–738

The International Journal of Advanced Manufacturing Technology

Development of an Advisory System for Trapped Material in Rapid Prototyping Parts

Ang Beng Yew, Chua Chee Kai and Du Zhaohui

School of Mechanical and Production Engineering, Nanyang Technological University, Singapore

Since the launch of the first commercial system in 1988, rapid prototyping (RP) technology is fast being accepted in the fields of product design and development, tool and die making, and recently, for biomedical applications. Although RP is able to produce objects of any complexities in theory, problems such as having material trapped inside parts with hollow features still occur frequently during normal operation. This could result in a distorted part being built, or the system being damaged. To reduce such occurrences, the purpose of this project was to develop an application program using the Unigraphics CAD/CAM system, to search automatically for the problematic features in a CAD file, and to provide the necessary modifications or recommendations. The initial focus has been on providing solutions for objects produced by stereolithography apparatus (SLA) and selective laser sintering (SLS). Provision has been made through the use of modular programming to allow more upgrades in future to other RP systems.

Keywords: Feature recognition; Rapid prototyping; Trapped material; Trapped volume; User function

1. Introduction

Rapid prototyping (RP) technology, otherwise known as solid freeform fabrication or layered manufacturing, refers to the physical modelling of a design using a special class of machine technology. RP systems are able to produce models and prototype parts quickly from 3D computer-aided design (CAD) model data, computer tomography (CT) and magnetic resonance imaging (MRI) scan data, and model data created from 3D object digitising systems [1]. RP systems join together liquid, solid, and powder to form physical objects, using an additive approach to building shapes.

Although RP offers many benefits compared to traditional prototyping methods, parts with complicated internal hollow features often have the problem of trapped waste material which cannot be removed from the part after building has been completed. This problem is especially significant in solid-based laminated object manufacturing (LOM) where sheets of paper are used in the fabricating process. In some cases, there is no feasible means of removing the trapped material without damaging the part. Even when removal of the trapped material is possible, great care has to be taken, leading to an increase in the overall postprocessing time. Liquid-based SLA and powder-based SLS have less of this problem as the materials they use can be drained out or vacuumed, but they do have the problem of parts being distorted owing to the presence of trapped material (also referred to as trapped volume) while building is in progress. The causes and effects of trapped material problems in parts produced by RP systems are detailed in an earlier work [2].

Current methods of working around the problem are by and large dependent on the experience and skill of the operators. To avoid the unnecessary loss of material, time and effort due to build failure, caused by the trapped material problem, an application program has been developed to analyse a CAD file for the problematic features, and to provide the necessary modifications and recommendations. The initial focus of the program is to provide solutions for simple parts made by SLA and SLS. However, the modularly designed program can be modified in future to detect more complicated features, and to analyse parts to be produced by other RP systems.

2. Design Considerations for the Trapped Material Advisory System (TMAS)

2.1 Functional Requirements

The trapped material problem, as the name implies, is caused by the presence of certain types of feature that trap the build material while building is in progress or when building is completed. Although the problem caused by these features can sometimes be removed by simply changing the orientation of the part to be built [2–5], a situation can arise where a drain hole must be incorporated into the original CAD object to remove the trap. Another remedy would be to split the part

Correspondence and offprint requests to: Dr Chua Chee Kai, School of Mechanical and Production Engineering, Nanyang Technological University, Nanyang Avenue, 639798, Singapore. E-mail: mckchua@ntu.edu.sg

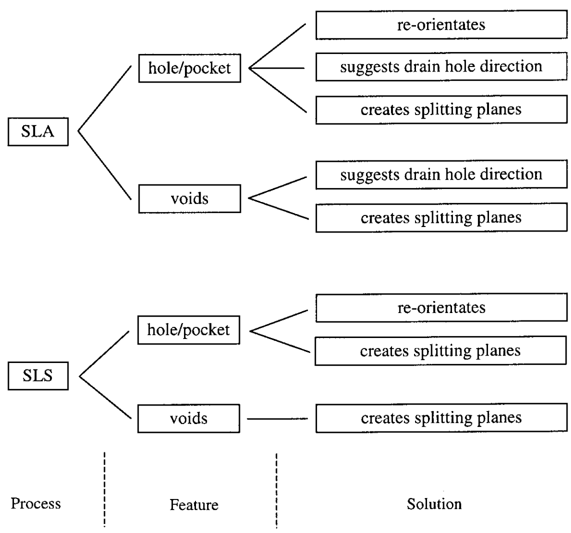

Fig. 1. Solution tree of TMAS.

along the axis of the problematic feature in the CAD object and rejoin the sections when the building is completed.

Taking the above into consideration, the trapped material advisory system (TMAS) has been designed to fulfil the following requirements:

1. Ability to detect a circular blind hole, rectangular pocket, cylindrical and rectangular voids in a CAD object that will be built by SLA and SLS.
2. Ability to re-orientate, suggest direction of drain hole and create splitting planes for a CAD object that will be built by SLA.
3. Ability to re-orientate and create splitting planes for a CAD object that will be built by SLS.

The relationship between the process, feature and required solution can be represented in the solution tree shown in Fig. 1.

2.2 Platform for Developing TMAS

The original intention was to develop TMAS as a stand-alone program that would be able to detect the problematic features in the STL (stereolithography) file of the object. However, an STL file consists only of an unordered list of triangular facets representing the outside skin of the object [6]. Topological information such as which are the facets that make up a particular feature, is unavailable. To create a system that is able to recognise features out of this unordered list of facets would be too time-consuming, and would probably require some form of user intervention [7]. Thus, it was decided to develop an application program using an existing CAD/CAM system, taking advantage of the topological information that exists in the CAD model, and the built-in functions that would facilitate feature recognition.

The platform chosen for developing TMAS is Unigraphics (UG), Version 13. The application programming interface (API) of this hybrid CAD modelling system is known as the UG/Open API (or user function) which is a C language interface. The header files are ANSI C compliant and also support the C++ language. UG/Open allows the user to model parts programmatically, query the object model, create assemblies, create drawings, and so on [8]. With the API, users can create custom applications using UG to suit their requirements.

2.3 Structural Design

Although the current emphasis is on SLA and SLS, trapped material remains a serious problem for objects that will be built by LOM or other RP systems. Hence, the subsequent development of TMAS will focus on providing solutions for LOM and these systems. To this end, the structural design of TMAS must be modularised to accommodate changes and upgrades in the future. Table 1 describes the functions of the individual modules that make up the TMAS. There are a total of four modules with the main module controlling the rest.

2.4 Functional Design

TMAS can be started once a CAD file is loaded into Unigraphics. The main module will first activate the *chkfile* function (see Fig. 2) to check if the object in the file is a solid object. Once this requirement and a few others have been satisfied, the main program will proceed to call the *detect* function which will prompt the user to select the system on which the part will be built. With this information, the *detect* function will be able to decide which features in the CAD object are likely to result in the trapped material problem. For SLA objects, blind holes and pockets with their openings facing upwards, together with internal voids, will be treated as potential problematic features. In the case of SLS, blind holes and pockets with their openings facing downwards, as well as internal voids, would be considered instead.

As it cycles through the features in the object, *detect* will capture important information such as the type, location, and size of the problematic features, and store them in an array structure created in the memory. Next, the true depth and maximum opening area of the trapped material in the problematic feature will be calculated. Take for example the blind hole in Fig. 3, the resin in the hole remains trapped until the SLA machine starts to build the opening where it is able to come into contact with the resin in the reservoir again. The true depth of this body of resin represents the maximum depth that the trap will experience and this is actually equal to depth d:

$$\text{True depth} = d = h \cos\theta \quad (1)$$

where h is the depth of the hole and θ is the angle made by the axis of the hole with the Z-axis.

Similarly, the maximum opening area of the trap resin is given by A_p which is:

$$\text{Maximum opening area} = A_p = A/(\cos\theta) \quad (2)$$

where A is the opening area of the hole.

The calculations for other features and features in SLS parts are based on a similar method.

Besides recording this information, the seriousness of the trapped material problem associated with each problematic

Table 1. Modules that make up TMAS.

Module (C file)	Functions in file	Primary tasks
Main.c	*ufusr*()	Controls flow of program
Chkfile.c	*chkfile*(p1, p2, ...)	Checks CAD file for solid object
Detect.c	*detect*(p1, p2, ...)	Detects and records problematic features, measures seriousness of trapped material problem
	convert(p1, p2 ...)	Converts parameters from text label to floating-point number
Modify.c	*modify*(p1, p2, ...)	Determines most problematic feature
	sla-remedies(p1, ...)	Re-orientates/drain holes/splitting planes
	sls_remedies(p1, ...)	Re-orientates/splitting planes/detects "growth"

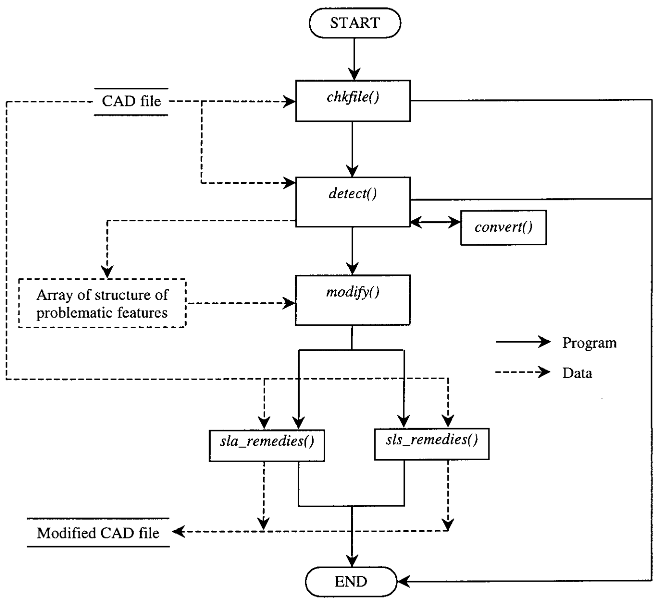

Fig. 2. Simplified flowchart of TMAS.

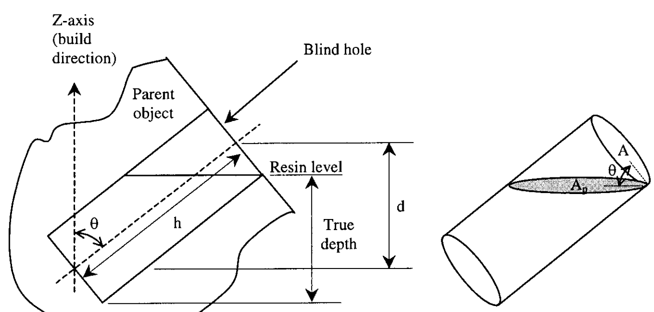

Fig. 3. Calculation of the true depth and opening area for the blind hole in an SLA part.

feature (see Eq. (3)) will be evaluated and summed to give the overall serious index of the trapped material problem of the object at this particular orientation. However, if the feature is a pocket or a rectangular void, the feature's serious index will be multiplied by a factor of 1.5 to reflect the higher tendency for such features to result in the trapped material problem.

$$\text{Serious Index} = \text{Depth weightage} \times \text{True depth} + \text{Area weightage} \times \text{Maximum opening area} \qquad (3)$$

Equation (3) shows that the greater the depth and surface area of the trapped material in the feature, the more serious the problem. Ideally, the depth and area weightage should be established through the building of test parts. Once the overall serious index has been calculated, the main program will proceed to activate the *modify* function. By comparing the index calculated by *detect* for each problematic feature, the *modify* function will determine which is the most problematic feature and proceed to either the *sla_remedies* or *sls_remedies* for modification. The flowcharts for these three functions are shown in Figs 4 and 5.

Using the *sla_remedies* as an example, the function will query the most problematic feature for its type (see Fig. 4). If it is a void, the control of the program will be passed to the algorithm that determines the direction of the drain hole and the location of the splitting planes (see Fig. 5). This algorithm makes use of the feature's face data to create bounding boxes, which will subsequently allow the drain hole's axis and splitting planes to be created and positioned correctly with respect to the feature. Referring to Fig. 4 again, if the feature is a blind hole or pocket (i.e. not a void), the function will proceed to calculate the angle (safe angle) at which this feature should be orientated with respect to the build direction (*Z*-axis). There are two ways to calculate the safe angle; one is based on the safe (maximum allowable) depth of the liquid trapped in the feature, and the other is based on the safe (maximum allowable) surface area of this pool of liquid. Both requirements are contradictory in the sense that to have a small depth, the safe angle should be large, whereas to have a small area, the safe angle has to be small. Therefore, the algorithm will compare both the depth weightage and area weightage assigned earlier in the detect module and decide which safe angle to adopt. Depending on which safe angle is adopted, its value can be determined from Eqs (1) or (2) by substituting the maximum allowable depth and h or the maximum allowable area and A, and solving for θ.

If the safe angle based on depth is adopted, the current angle will be considered unsafe if it is smaller than the safe angle (i.e. greater depth). However, if the safe angle based on area is adopted, the current angle will be considered unsafe if it is larger than the safe angle (i.e. greater area). If either of these happens, the feature (and hence the part) will be rotated five times. The first is a complete flip-over

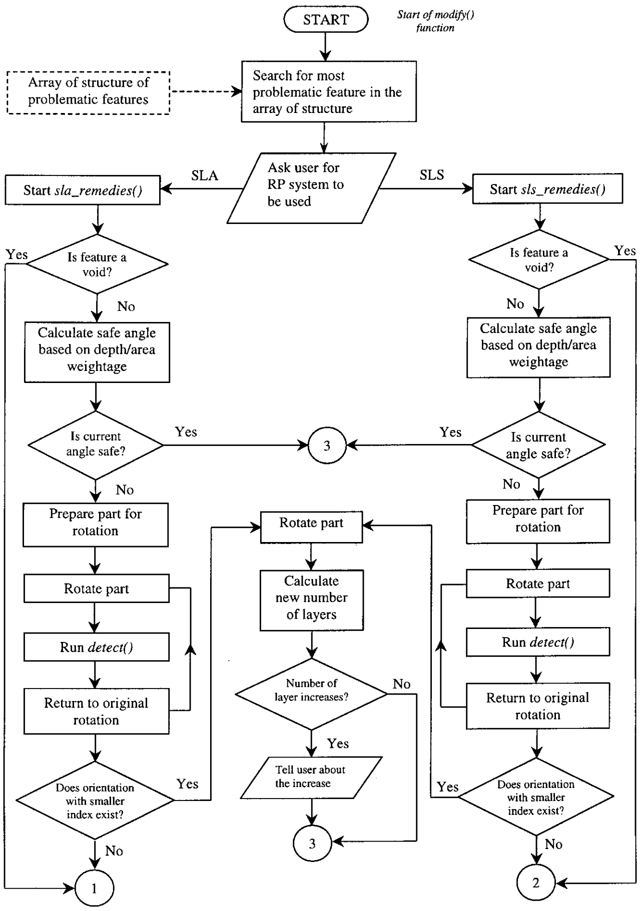

Fig. 4. Flowchart of the modification module (reorientation).

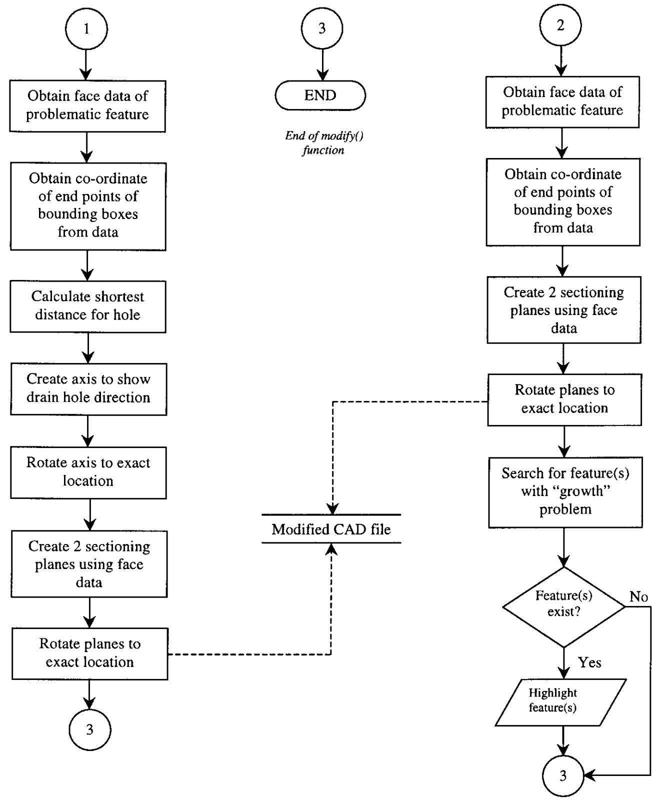

Fig. 5. Flowchart of the modification module (drain hole/splitting planes).

and provided this does not result in more problematic features, it will be the preferred orientation as the number of layers to build the object remains the same. Four more rotations based on the difference between the current angle and safe angle will be carried out: anticlockwise about positive and negative *X*-axis, and anticlockwise about positive and negative *Y*-axis. As there can be an infinite number of axes about which the part can be rotated, those used by TMAS were already present and need not be specifically defined. After each rotation, the function *detect* will be called upon to calculate the overall serious index of trapped material. If the smallest index of the five orientations is smaller than the index of the original orientation, then the object will be rotated. Any increase in the number of layers will also be reported to the user. If a better orientation cannot be found, the algorithm to create the drain hole axis and splitting planes will be activated to provide the user with an alternative solution. The operation of *sls_remedies* is similar to *sla_remedies* except that there will be no suggestion for the creation of drain hole. However, the *sls_remedies* has a routine to check for the existence of the "growth" problem in features with small cross-section [2].

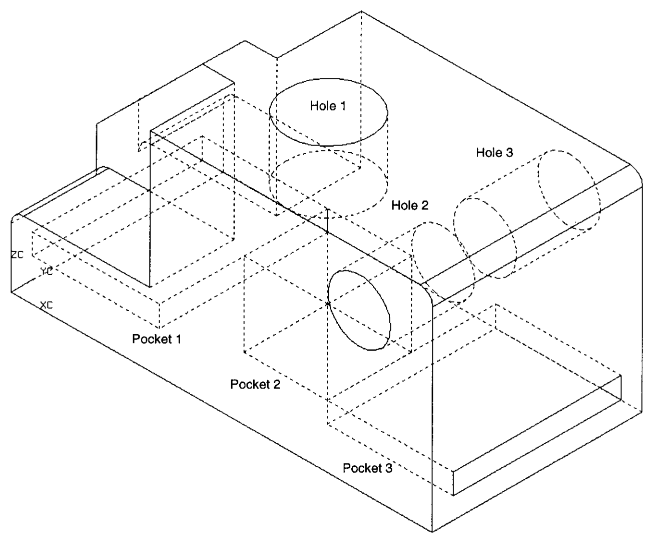

Fig. 6. Sample object used to demonstrate the operation of TMAS.

3. Sample Run

To illustrate the operation of TMAS on an SLA part, a sample object with trapped material features in different directions is created and analysed using TMAS (see Fig. 6). There are three blind holes and three rectangular pockets placed in such a way that a simple re-orientation of the object would not be able to

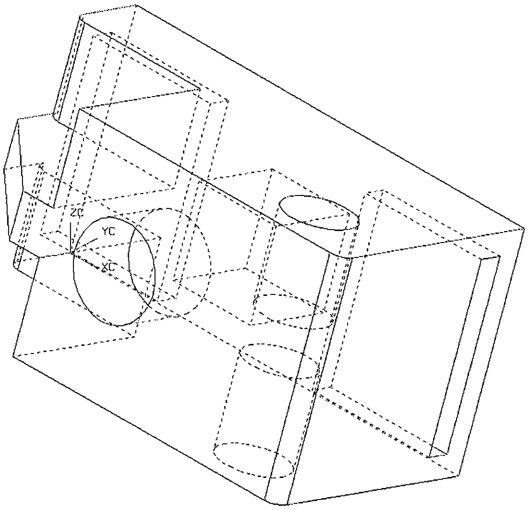

Fig. 7. The object being rotated to search for a safe orientation.

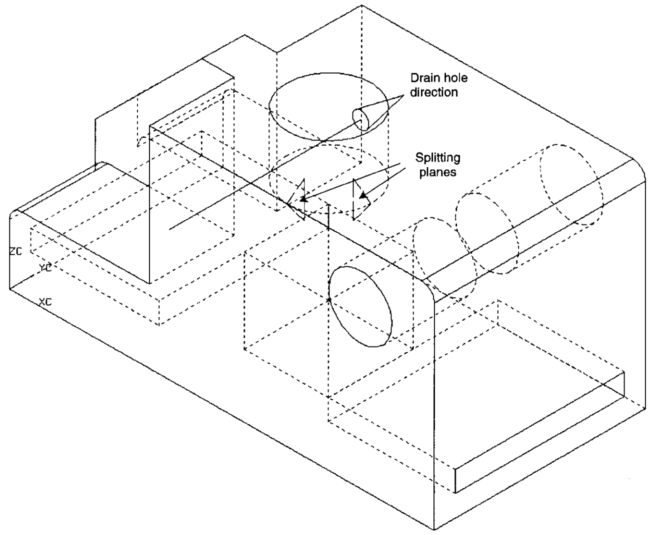

Fig. 8. Drain hole direction and splitting planes suggested by TMAS.

solve the trapped material problem. The safe depth of feature is set at 5 mm which is obtained from test parts built previously using SL5170 epoxy resin on an SLA 250. None of these features has an opening area greater than the safe area set at 1600 mm^2. The depth and area weightages are set at 80% and 20%, respectively. At the original orientation shown in Fig. 6, the problematic feature is Hole 1 with diameter 20 mm and depth 15 mm.

When TMAS starts running, information regarding the hole is being recorded and once it is determined that the safe depth has been exceeded, the modify module will calculate the safe angle at which the object should be orientated. Since the depth weightage is larger, Eq. (1) is used to determine the safe angle and it is found to be 70.53°. Five rotations (one of which is shown in Fig. 7) are subsequently carried out based on the difference between the current angle and the safe angle to determine whether any safe orientation exists.

For each orientation, the *detect* function is called upon to calculate the overall serious index of trapped material problem. These results are summarised in Table 2. Compared to the original orientation, none of the proposed orientations is able to give a smaller serious index and hence the TMAS proceeds to calculate the direction of axis for the drain hole and create the splitting planes (see Fig. 8). The final decision is left to the user to adopt either of these suggestions.

Table 2. Summary of overall serious index for various orientations.

Type of orientation	Overall serious index	Problematic features
Original	74.832	Hole 1
Flip-over	144.000	Pocket 2
About positive *X*-axis	245.065	Hole 1, Hole 3
About negative *X*-axis	245.065	Hole 1, Hole 2
About positive *Y*-axis	290.074	Hole 1, Pocket 1
About negative *Y*-axis	290.074	Hole 1, Pocket 3

4. Conclusion

Trapped material is a serious problem that could result in build failure in several RP systems. TMAS is currently able to detect some common features that cause the trapped material problem in an object to be produced by SLA or SLS. When a simple re-orientation based on consideration for the depth and area of the trapped material is unable to solve the problem, alternative solutions such as building drain holes and splitting the part will be proposed. The work up to this stage has been on building a framework on which further improvements can be made in future. These include:

1. Ability to analyse more complicated trapped material features, preferably freeform features.
2. Ability to provide solutions for multiple trapped material features besides the most problematic one.
3. Ability to analyse LOM objects.
4. More reliable prediction on build failure using data obtained from real test parts.
5. An improved algorithm to determine the best orientation, taking into account the accuracy of part, type of material used, and the amount of support structure needed.

Of particular interest is the last point, as these are the usual concerns besides the trapped material problem, when building an object using RP systems. By taking these into consideration in the future development of TMAS, the system could evolve into a truly useful tool that helps a user to decide the best way to build an RP object.

References

1. T. Wohlers, Rapid Prototyping State of the Industry: 1997 Worldwide Progress Report, Rapid Prototyping Association of the Society of Manufacturing Engineers, Michigan, 1997.

2. B. Y. Ang, C. K. Chua and Z. H. Du, "Study of trapped material in rapid prototyping parts", The International Journal of Advanced Manufacturing Technology (accepted), 1998.
3. P. F. Jacobs, Rapid Prototyping and Manufacturing: Fundamentals of Stereolithography, 1st Edn, Society of Mechanical Engineers, pp. 14, 15, 160–163, 180–183, 1992.
4. CAD/CAM Publishing, "Trapped volumes", Rapid Prototyping Report, 6(2), pp. 6, 1996.
5. M. Durham and T. Grimm, "SLS and SLA: Different technologies for different applications", http://www.acceltechinc.com/slssla0596.html, 1996.
6. C. K. Chua and K. F. Leong, Rapid Prototyping: Principles and Applications in Manufacturing, John Wiley, Singapore, 1997.
7. R. I. Campbell, "Optimisation of build orientation for feature-based surface roughness requirements", Proceeding of the First International Conference on Rapid Prototyping and Manufacturing '98, Beijing, pp. 632–638, 1998.
8. Electronic Data Systems Corporation, Unigraphics Divisions, "User Function System Overview", UG On-line Documentation, pp. 1-1–1-3, 1995.