A geometric aid during the first stages of product collaborative design

Denis Pallez, Parisa Ghodous, M. Martinez

A GEOMETRIC AID DURING THE FIRST STAGES OF PRODUCT COLLABORATIVE DESIGN

Denis PALLEZ, Parisa GHODOUS, Michel MARTINEZ

PRISMa Laboratory – University of Lyon1 – France

pallez@netcourrier.com

ABSTRACT: (196 words)

The technological advances carried out these last years in the field of products development led the researchers to elaborate the approaches that reduce the cost and time of product development, enhance the quality of product and help the designers to be more creative. These objectives are difficult to obtain due to a large number of phases, which should be carried out during the product development and the large number of experts of different disciplines that are involved. Currently, computer aided systems and software have concentrated on the capture and representation of geometrical shape and technical information as opposed to providing supports for product design in the earlier stages of design process. In a non–routine design, it is delicate and extremely complex to obtain the best products answering customer's specifications. The aim of this paper is to present a methodology to assist designers during the first stages of design. The objective is not to construct automatically the shape but both to automate a certain number of heavy and tiresome tasks, and assist designers during collaborative design. In best case, this assistance makes the designers' stimulation possible by presenting them solutions that they had not thought before.

ARTICLE: (3381 Words without references and figures)

The will to capitalize the know-how of firms and to reduce times of production make that one now truly approaches the CAD/CAM systems in the optics of a functional/conceptual modeling (Minich and Pallez 1999). The development of CAD/CAM systems knew several phases; restricted themselves to geometry first, they was little by little enriched with information of higher semantic level; those information could be dedicated to specific trades that took part in the product design. By this way, features and product modeling represent an improvement of design models by adding to the geometry necessary information to the manufacture, for example.

The goal is to assist a CAD software user during the earlier stages of the product design. However the current systems are still based on geometry and in order to achieve the wished goal, it is necessary to delay computations and to introduce higher semantic level concepts (Minich and Pallez 1999). Introducing form features carried out a first effort (Salomons 1994). They assemble elements of geometry of very low level (as faces or edges) to form generic entities easily handled by the engineer because they could directly be associated with functionality (Feng, Huang, et al. 1996) like sliding motion for a groove or buttress for a shouldering wall. Even if the engineer is brought to manipulate entities whose he apprehends, he is still obliged to think and to generate the product's shape to be designed. Moreover form features are essentially based on geometry as it is made by current software systems. But these ones give a too significant part to the geometrical models by encapsulating them with specific information of various activities. And it is to go against a current tendency of the research that tightens to oust geometry of its central position (Brun 1997). Research tends to reverse this inclination: it makes possible to represent the product from a more conceptual point of view in introducing functions. The latter represent the translation of the product's specifications from the first stages of the design (that are the most determining). To design a product that satisfies all its functions makes possible to obtain a product of quality, taking into account the cost, the longevity and the adaptation (Ullman 1997). Currently, only manual or assistance techniques of functions product's development exist (value analysis, Qualitative Function Deployment (Ullman 1997), FAST...). Handled information is mainly expressed in natural language (Figure 1a), which makes it not easily automatisable even if there exists models making possible to build a functional decomposition facilitating the product's simulation. FBS (Function Behaviour Structure (Tomiyama, Umeda, et al. 1993) (Umeda, Ishii, et al. 1996) (Ranta, Mäntylä, et al. 1996) describes the product according to three levels: the first draws up functions (Figure 1b); the second specifies how to fulfill these functions by behaviors (Figure 1c) ; the last level describes behaviors like a sequence of states of the product's components (Figure 1d). Consequently, the capacity to treat functions by computer opens the way with an automation of the earlier stages of design.

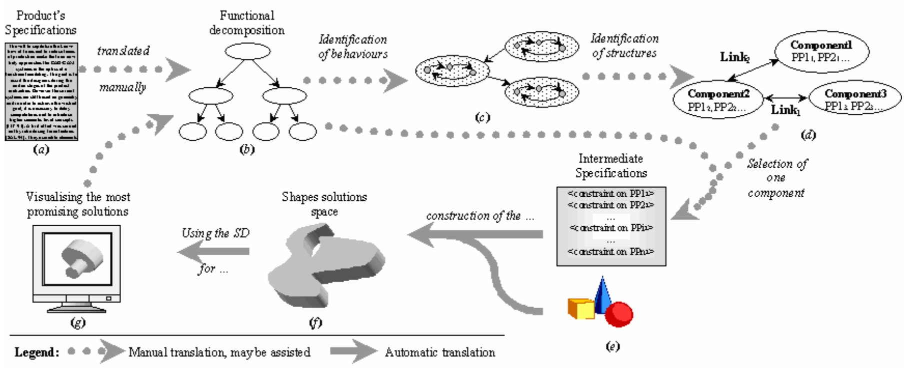

Figure 1: Overall architecture of the system

However, in a more general way, one notes that methods of assistance in manufacturers products design are manual during the first phases of design because manipulated information are mainly written in natural language. As a consequence, any automation attempt will be difficult because, at this time, one knows that it is computerly and automatically hard to interpret the significance of a text. Therefore, it limits the aid proposed by the different software tools; they are more used in a verification level, as simulators for instance. So, we assume that those information can be manually translated into more easily interpretable information by a software system: the translated information are mathematical constraints on parameters of a relatively elevated semantic level (the volume of the design object, the coefficient of penetration in air and so on… Figure 1e). Those parameters are called *intermediate parameters* and are defined as quantifiable and measurable entities referring to the physical world and are not necessarily related to geometry. The set of constraints using intermediate parameters defines what we call the *intermediate specifications* (Gardan, Minich, et al. 1999a) (Gardan, Minich, et al. 1999b). It could appear abusive to suppose that the initial specifications (Figure 1a), expressed in natural language, can be translated into intermediate specifications (Figure 1e). However, in some number of cases, the thing is really possible and by this way, a starting point is obtained for the almost automatic shape synthesis. For example, in the case of a box design, aesthetics functions may be converted as the following: the ratio of length to height of the box approaches the gold number $\left((\sqrt{5}+1)/2\right)$. The corresponding constraints would be length = $\left((\sqrt{5}+1)/2\right)$ /height. In a larger extent, the handling of a water bottle by a human being, which is a function, may be converted into constraints on the weight, the compactness and so on, which are parameters. In (Gardan, Minich, et al. 1999b), we have defined an intermediate constraint by a quadruple <*IP*, *R*, *Exp*, *W*> where *IP* is an intermediate parameter, *R* is a relation among $\{<, >, =, \neq\}$ that must be considered as fuzzy relation, *Exp* is an arithmetic expression and *W* is the relative weight of the intermediate constraint in comparison with the other constraints of the intermediate specifications. For the moment, only one designer gives the weight for all the intermediate constraints.

From intermediate constraints and a library of primitive shapes (Figure 1e), we propose to size every shape of the library so that they verify the intermediate constraints (Gardan, Minich, et al. 1999a). We obtain what we called the solutions space, which contains all solutions (Figure 1f). We agree that a shape is defined by what we call *terminal parameters* (mainly geometric and of weak semantic level: radius, length, width…). We suppose, in addition, that an expert of the design domain has provided the software system with sufficient knowledge so as to know how to translate an intermediate constraint into terminal constraints. The latter apply to terminal parameters of a parameterized shape of the library. For example, a weak penetration coefficient can result in a weak radius for the circle or a weak width and an elevated length for the rectangle whether the circle and the rectangle are parameterized shapes of the library… As the shape synthesis of primitive shapes is not very interesting from an industrial view point, we have studied different methods permitting to synthesize more complex shapes (Gardan, Minich, et al. 2000).

As we assume that terminal parameters vary in real intervals, there is a great number of shape solutions that satisfied the intermediate constraints, or even an infinite number. However, the conviviality of a shape synthesis software tool implies the presentation of a restricted number of solutions to designers. It means that it is necessary to define a method searching for the best solutions among the synthesized shape solutions; it raises two problems: to browse the set of shape solutions in intelligent manner and to compare solutions between them.

The second problem consists in defining a degree with which a solution satisfies the intermediate specifications. This degree is called *Satisfaction Degree* and is obtained by computing a weighted average on satisfaction degrees of intermediate constraints affected of their weight. The computation of the satisfaction degree of an intermediate constraint (cf. Figure 2) is computed from the three following information (Gardan, Minich, et al. 1999a):

- the value of the intermediate parameter for the considered solution, so-called *real value*,;

- the *wanted value*, that corresponds to the value of the mathematical expression contained in the intermediate constraint;

- a curve depending on the mathematical relation used in the intermediate constraint.

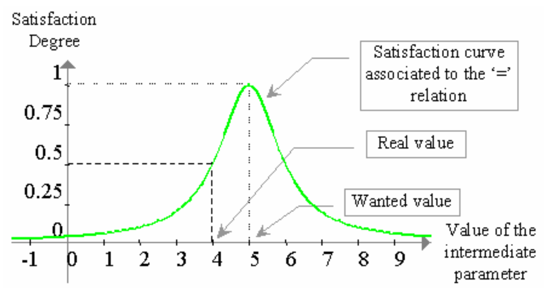

Figure 2: Satisfaction curve associated to the '=' relation for the following intermediate constraint: < IP, =, 5>

In fact, the satisfaction degree of an intermediate constraint is obtained by computing the existing gap between the real value and the wanted value on the considered curve. The computation of the intermediate specifications' satisfaction degree for a given solution is an operation called *estimation*.

We showed that the estimation of combined shapes (represented by a Boolean combination of primitive shapes contained in the library ≈ CSG tree) could not be deduced from the evaluation of the primitive shapes used to define combined shapes (Gardan, Minich, et al. 1999b).

Knowing how to estimate a shape solution, now we are able to browse solutions space that is modeled by variation intervals of terminal parameters for every parameterized shape of the library. As we have noticed that it seems difficult to use an exact optimization method that are mainly expressed in a mathematical way (as the Simplex method for instance), we study the possibility to apply some stochastic methods as simulated annealing method or genetic algorithms (Gardan, Minich, et al. 1999a). Then, we have proposed a method more adapted to our approach (Gardan, Minich, et al. 1999b). The latter consists first in sampling terminal parameter variation intervals. Secondly, each sample must be estimated in order to interpolate a curve in the case where the shape is defined by only one terminal parameter, a surface in the case where the shape is defined by two terminal parameters or a hyper surface in the case where the shape is defined by more than two terminal parameters. Afterwards, it is possible to determine with a mathematical method the maximum of the hyper surface. Moreover, the best solutions are obtained by sampling over again close to a certain number of maximum. Finally, shapes obtained by the application of the previous method can be presented to designers (Figure 1g). Once this step is finished, designers have to give their opinion on the selected solutions. In the case where there are not pleased with proposed shapes, and it will be often, designers have to modify intermediate specifications, modify the intermediate constraints weights, modify initial specifications of the product, add new intermediate constraints or new parameterized shape in the library, and so on…

In summary, the above methodology consists in translating manually functional information in constraints on physical entities of the product to design. From this information represented by the intermediate specifications, also from a library of parameterized shapes and expert knowledge, numerous shapes solutions are synthesized. To encourage the conviviality of the software system, the most promising solutions are searched and presented to designers so as to stimulate their creativeness.

We studied the validity of our methodology in the case of a very precise domain: foundry mould design (Gardan, Lanuel, et al. 2001). From an industrial viewpoint, the caster (foundry mould designer) cannot take the liberty to study and to estimate a big number of solutions in so far as the estimation of each solution is a long time consuming. Therefore, the caster uses trade rules to limit the solutions space. On the contrary, our methodology synthesizes too many solutions. So, in that study, we have modified our methodology by introducing some trade rules coming from an expert of the domain of foundry. The aim of the modifications was to reduce reasonably the solutions space by determining a priori the most promising solutions families, but by preserving an area large enough in order to preserve the property of creativity. For instance, the placement of pieces to manufacture makes the solutions space browsing difficult. So, by automatically computing the different possible arrangements of pieces in the mould in using including shape for pieces, it is possible to automatically define *classes of solutions*. This computation is less time consuming than testing and estimating each placement of pieces in the mould. Finally, the application of this method leads to a mould that the weight ratio is better of 40% than the one design by the caster.

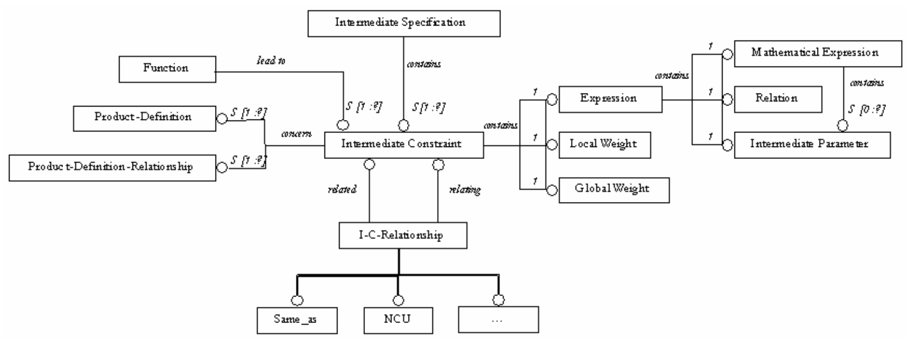

Figure 3: Intermediate specifications model expressed in an EXPRESS–G format

Recently, we worked on the way to facilitate a collaborative function-to-form mapping. Firstly, we have studied the problem of representation of experts' multiple-view in a collaborative conceptual design environment (Pallez, Dartigues, et al. 2001). In that study, we have defined some coherence rules between different models. Each expert at any time can define his model and may collaborate with the other models using a STEP standard language named EXPRESS–G (cf. Figure 3). As a consequence, when one model is manipulated, corresponding effects should be made automatically in the others. Finally, in (Pallez, Dartigues, et al. 2002), we have improved the methodology by studying the case of a water bottle design with three different design domains: experts on materials, experts on geometry and manufacturers. The resulting methodology is a function-to-form mapping in a collaborative context and is the following:

First step : Each design domain has to define its own intermediate specifications for only one component of the product to design. The intermediate constraints are deduced from the functional decomposition of the design product regardless of the other design domains participating in the design process. As a consequence, in this step, intermediate specifications of a domain will use only physical parameters of the domain. By this way, a designer of a domain can be considered as an expert of this domain in contrary of the methodology presented previously.
Second step : Next, as we are convinced that there exist relationships between constraints from one design domain to another, this step consists in establishing those constraint's relations. There is several ways to achieve this: either manually or semi–automatically by considering the rule "if two constraints from two different domains are deduced from the same function, then they are related each other".
Third step : This step corresponds to the solutions space generation and it is almost the same as our previous method. For the moment, we assume that only experts of design domain who have a shapes library are in charge of proposing solutions by applying the methodology presented previously.
Fourth step : Once shapes solutions are generated, experts of design domain who do not have a library must react to the proposed solutions by participating in the selection of the most promising solutions. So, in this part, experts from all design domains participate in selecting solutions.
Fifth step : As it is inconceivable that a promising solution could be find after the first try of shapes solution generation and selection, designers will be obliged to collaborate in order to modify the weights of constraints and/or add other intermediate constraints and/or add other parameterized shape in their library so as to increase the satisfaction degree of promising solutions. The function-to-form mapping process starts again from the second step until a promising solution satisfies all the designers involved in the product design.

Our experience shows us that an automatic and direct mapping of the functional information to geometric information represents a very difficulty problem for the moment. Moreover, numerous authors are working on this computer-aided-design problem (Gorti and Sriram 1996, Ranta, Mäntylä, et al. 1996, Rosenman and Gero 1996a, Tomiyama, Umeda, et al. 1993, Umeda and Tomiyama 1997, Zhihui and Johnson 1997). To synthesize our methodology, we have previously proposed to establish a median difficulty level that is represented by the intermediate specifications. We assume that the latter is manually obtained from initial specifications mainly expressed in natural language. We concentrate on the almost automatic mapping of the previous model into one or several shapes. The intermediate specifications model is made up of a set of constraints named intermediate; each constraint is made up of physical parameter also named intermediate. These parameters are quantifiable quantities that remain to a high semantic level. Our approach is independent of design domains even though experts of design domains must define some information, necessary to a good working of the function to form mapping. However, the notion of "perimeter", for instance, remains the same whatever the design domain considered. By this way, there is knowledge capitalization and as and when designs are done, one becomes less and less necessary to consult an expert of the considered domain. Information provided by designers permit, among others, the mapping of intermediate constraints into variation intervals of terminal (geometric) parameters. A Cartesian product of intervals defines the solutions space. On one hand, we have assumed that the shapes contained in the library were primitive shapes. We can show that the addition of more elaborate shapes, or combinations of primitive's shapes, don't modify the proposed approach. However, the number of terminal parameters for those shapes increases considerably.

Our future works are numerous. First of all, in short-term, it is necessary to identify all the possible relations between constraints from different design domains in order to allow more precise communication between experts of these different domains. Then, it is important to study how to maintain the consistency of models. In that case, future works will focus on the definition and formalization of coherence rules between different models so as to improve proposed multiple-view model. Secondly, the application of our methodology in the very precise framework of foundry mould design permitted us to consider the automatic creation of shapes that would be going to enrich the library. An evolution of our methodology would consist therefore not to preserve shapes in a library but to construct these shapes automatically according to concerned design domains. The idea would be to elaborate a second intermediate model, between the intermediate specifications and the solutions space, what would permit a less abrupt passage again between functions and shapes. Moreover, according to our methodology, the estimation operation requires instantiation of a solution that is the assignment of a real value to every terminal parameter of the solution. Another improvement would consist in estimating a set of solutions rather than a unique solution: for instance, estimate the shape “circle” without knowing precisely the value of the radius. The shape “circle” is called class of solution. The possibility to estimate a class rather than a shape would permit, for example, to construct first the satisfactory classes, of which the better would be examined. Then, in the most promising classes, one would choose the most promising solutions.

In long term, future works should be related to geometric reasoning: it could be interesting for the experts to define intermediate constraints using other kind of relation. For instance, instead of using well-known mathematical relations ($\leq$, $\geq$, $=$, $\neq$), experts on geometry would like to use a “look like” relation ($\equiv$) in order to introduce new experiences on shapes. Once it will be done, it will be very interesting for designers, and especially for experts who have a shape library (expert on geometry), to combine shapes contained in the library so as to be more creative. For instance, evolutionary algorithms could be used (Taura, Nagasaka, et al. 1998) (Rosenman and Gero 1996b) (Gardan, Minich, et al. 2000).

Perspectives in a more general context are also numerous and often make call to other domains of research. In particular, works done in artificial intelligence could serve basis for a better semantics understanding of functional information. It would allow software system to provide a more precocious aid for the function-to-form mapping. In addition, the quality of the man-machine interface is an essential notion for appreciated software by its users. Even if improvements are brought to software which implements the function-to-form mapping, the place of designers is still very important in this mapping. So, a scrolling of the most promising solutions should be done so as to give designers the possibility to intervene when an aspect of the shape solutions suits them or displeases them. It raises important difficulties of zones designation by designers, of their interpretation from a functional point of view and how they will speak in the next step of function-to-form mapping.

If solutions follow each other in any order, the operation will be especially long and laborious for designers. To make it convivial, we foresee to present solutions so that they present a geometric continuity; by this way, their scrolling will appear like an animation. It presents the advantage to make evolve solutions in a progressive manner. To provide this geometric continuity, it is necessary to realize an algorithm that browse the solutions space and find a solution that looks like another. Another possibility is to realize a geometric morphing algorithm that converts progressively a solution into the following solution. The drawback is to generate shapes that are not solutions.

Biographical notes:

Denis PALLEZ is currently an associate professor in the department of computer sciences at the University of Lyon1, France, in the PRISMa laboratory. He obtained his PhD in computer sciences in January 2000 at the University of Metz (France). He also spent four months as an invited researcher at the institute of technology at Montreal (ETS). His research interest is in the area of CAD/CAM including conceptual and functional design, needs and functions modeling, intelligent CAD systems, shape synthesis, technical data management system.

(83 words)

Parisa Ghodous is currently associate professor in computer science department of University of Lyon I. She is head of technical modeling team of LIGIM (Laboratory of Computer Graphics, Image and CAD/CAM Modeling). Her research expertise is in the following areas : Standards, Product data exchange and modelling, Design Process Modelling, Collaborative Modelling, Distributed Technical Data Base and Interoperability. She is in editorial boards of two journals and in the committees of many relevant international associations such as concurrent engineering, ISPE, ECPPM, Interoperability.

(82 words)

Michel T. Martinez is a Professor in the Engineering Department of the Institute of technology, the University of Lyon (France) and vice-head of the Advanced Manufacturing Centre (AIP) of Lyon. After a PhD degree in Computer System Technology, he conducted research on Object-oriented application development for industrial applications, CIM control system design, Assembly Process Generation and Group Technology. Since 1999, his current research interest focuses mainly on Concurrent Engineering and New factory Organisations. Actual main projects include technical data exchange and sharing, collaborative integrated development and distributed architecture for technical information technology.

(92 words)

References: (460 Words)

Brun, J.-M. 1997. "Modèle produit : les conditions de cohérence et leur évolution au cours du processus de conception." Revue internationale de CFAO et d'Informatique graphique 12(5): 513-529.

Feng, C. X., Huang, C. C., Kusiak, A. and Li, P. G. 1996. "Representation of functions and features in detail design." Computer-Aided Design 28(12): 961-971.

Gardan, Y., Lanuel, Y., Pallez, D. and Vexo, F. 2001. "A methodology for a function-to-shape translation tool in foundry." Computers in Industry 44(2): 117-130.

Gardan, Y., Minich, C. and Pallez, D. 1999a. On shape to specifications adequacy. IEEE Information Visualisation (IV), London, Great Britain, D. Dornfeld (eds), Kluwer academic.

Gardan, Y., Minich, C., Pallez, D. and Perrin, E. 1999b. From Functions to Shapes. Third International conference on Engineering Design and Automation (EDA), Vancouver, Canada.

Gardan, Y., Minich, C., Pallez, D. and Perrin, E. 2000. Towards a specifications-to-shape translation tool. Third International Symposium on Tools and Methods of Competitive Engineering (TMCE), Delft, The Netherlands.

Gorti, S. R. and Sriram, R. D. 1996. "From symbol to form: a framework for conceptual design." Computer-Aided Design 28(11): 853-870.

Minich, C. and Pallez, D. 1999. "Vers des outils informatiques d'assistance aux phases amont de la conception - Etat de l'art." Revue Internationale de CFAO et d'Informatique Graphique 14(2): 177-206.

Pallez, D., Dartigues, C. and Ghodous, P. 2002. Towards a Collaborative Function-to-Form Mapping. Concurrent Engineering, Cranfield University, UK.

Pallez, D., Dartigues, C., Ghodous, P. and Martinez, M. 2001. Data Architecture for Collaborative Conceptual Design. 8th IEEE International Conference on Emerging Technologies and Factory Automation, Antibes (France).

Ranta, M., Mäntylä, M., Umeda, Y. and Tomiyama, T. 1996. "Integration of Functionnal and Feature-based product modeling - the IMS/GNOSIS experience." Computer-Aided Design 28(5): 371-381.

Rosenman, M. A. and Gero, J. S. 1996a. "Modelling multiple views of design objects in a collaborative CAD environment." Computer-Aided Design 28(3): 193-205.

Rosenman, M. A. and Gero, J. S. 1996b. "Modelling multiple views of design objects in a collaborative CAD environnement." Computer-Aided Design 28(3): 193-205.

Salomons, O. 1994. Computer support in the design of mechanical products. Constraint specification and satisfaction in feature based design for manufacturing, University of Twente.

Taura, T., Nagasaka, I. and Yamagishi, A. 1998. "Application of evolutionary programming to shape design." Computer-Aided Design 30(1): 29-35.

Tomiyama, T., Umeda, Y. and Yoshikawa, H. 1993. A CAD for functionnal Design. Annals of CIRP'93.

Ullman, D. G. 1997. The mechanical design process, 2nd edition: McGraw-Hill.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y. and Tomiyama, T. 1996. Supporting conceptual design based on the function-behavior-state modeler. Artificial Intelligence for Enginnering Design, Analysis and Manufacturing.

Umeda, Y. and Tomiyama, T. 1997. "Functional reasoning in design." IEEE Expert Intelligent Systems & Their Applications 12(2): 42-48.

Zhihui, Y. and Johnson, A. L. 1997. "On estimating the feasible solution space of design." Computer-Aided Design 29(9): 649-655.