Data Architecture for Collaborative Conceptual Design

Denis Pallez, Christel Dartigues, Parisa Ghodous, Michel Martinez

Data Architecture for Collaborative Conceptual Design

Denis Pallez
Christel Dartigues
Parisa Ghodous
PRISMa, LIGIM – University of Lyon 1
43, boulevard du 11 novembre 1918
69622 Villeurbanne Cedex
France

Michel Martinez
PRISMa – INSA of Lyon
20, Avenue Albert Einstein
69621 Villeurbanne Cedex
FRANCE

***Abstract* – Currently, computer aided systems have concentrated on the capture and representation of geometric shape and technical information, as opposed to providing support for product design. The aim of this paper is to propose a framework for an integrated software that assists designers in the early design phases to work in co-operative and collaborative manner.**

***Keywords* –** Computer Aided Design, Shape synthesis, Multiple View Points, Product Data Integration, Concurrent engineering.

I. INTRODUCTION

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 the great number of phases, which should be carried out during the product development (Fig. 1) and the important number of experts of different disciplines that are involved. Concurrent Engineering is considered one of the key concepts that enables the companies to reach these objectives [1].

Concurrent engineering is a wide field of research. The researchers who are interested in concurrent engineering, work on different aspects such as [2]:

- Philosophical aspect deals with the boundaries of the responsibility and the authority, culture and organization management,
- Methodological aspect deals with system thinking, approaches to system complexity, systems engineering, product realization taxonomy and system integration,
- Conceptual aspect deals with concurrency and simultaneity, modes of concurrency and cooperation, work flow mapping and
- Virtual aspect deals with capturing life cycle intent and information modeling.

Our work is situated on the virtual aspect. In the product design, each expert works with his own applications, handles his own data, has knowledge and constraints which are specific to his field of work and has his own point of view on the product [3]. This heterogeneity implies many problems, in particular the exchange and the sharing of information with the other experts. It is thus necessary to make it possible to each expert to represent adequately its data while facilitating integration and communication of their data with the other experts.

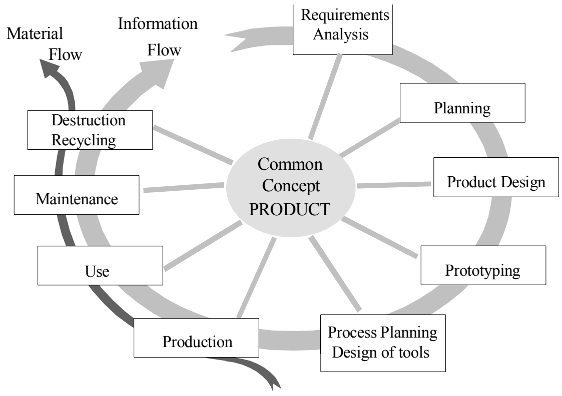

Fig. 1 Different viewpoints on product development

In this context, our objective is to develop a concurrent and collaborative system allowing experts to participate on product development as soon as possible and help them to work together [4].
In the following, firstly, we present the function to form mapping approach as a conceptual design technique. Then we explain why it is more realistic to consider this methodology in a collaborative context and we present enhanced version of the function to form mapping approach presented in [5,6]. Finally, we present the data models that are necessary to achieve a collaborative and conceptual design aided system based on this methodology. Due to the reasons of normalization, these models are described by EXPRESS-G formalism [7].

II. THE FUNCTION TO FORM MAPPING AS A TECHNIQUE FOR CONCEPTUAL DESIGN

The aim of any design is to obtain as soon as possible the product so as to be most profitable as possible. In a non–routine design, it is delicate and extremely complex to obtain the best products answering customer's specifications. Indeed, a great number of different experts of different disciplines participate in the product design. The difficulty lies in the fact that those persons have to collaborate. Shape is today the main representation of a product, even though the current trend is to remove geometry from its central position in order to add high level information. In fact, geometry constitutes the starting point of many activities such as mechanical optimizations, kinematics simulations, and so on. The function to form mapping appears to be one of the most important activities of the design process and up till now happened manually.

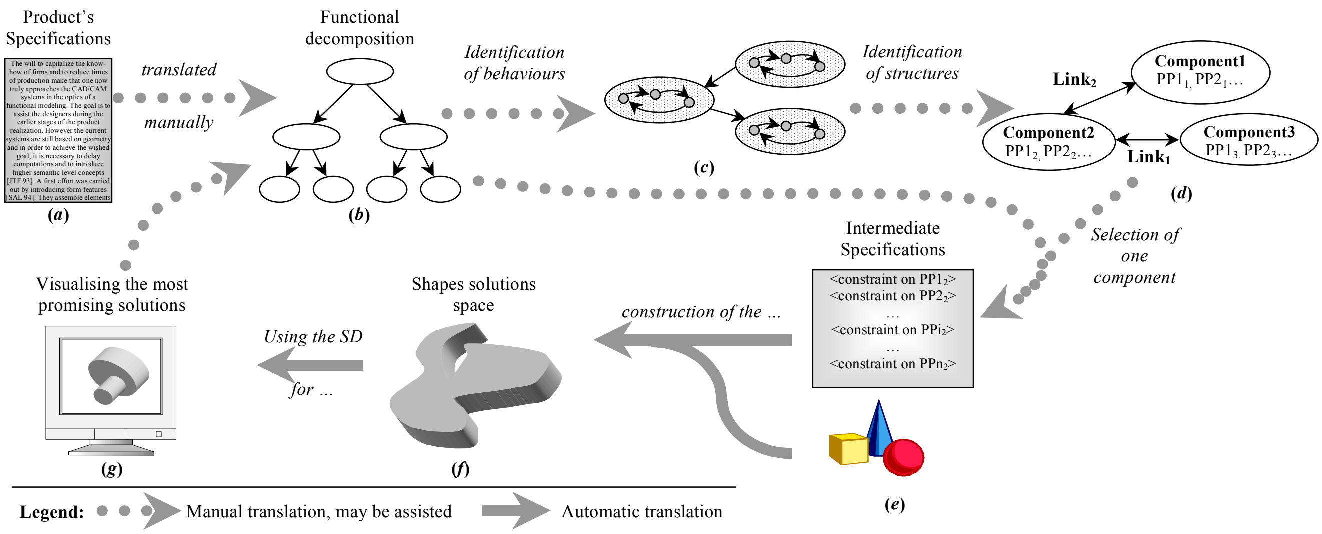

Fig. 2 Overall architecture of the system

This activity is very important both for the choices that are made and for the quantity of work that it represents [8]. That is why we have the desire to assist designers in the first stages of product design. The objective is not to construct the shape automatically but to both automate a certain number of heavy and tiresome tasks, and assist the designer during the first stages of design. In best case, this assistance makes the designers' stimulation possible by presenting them solutions that they had not thought before.

We suppose that the first information we have at our disposal are included in the customer's specifications which is expressed in natural language (Fig. 2a). Due to the functional decomposition (Fig. 2b) using functions of making, maintaining, prevention, control [9] and allowing [10], behaviors of the product can be identified (Fig. 2c) in order to respect the philosophy of FBS [11, 12, 13]. Next, with the help of the same philosophy, structure of the product can be obtained (Fig. 2d). This model can be identified with product's components linked by assembly relation.

In some cases, we have noticed that a function can also be decomposed into a set of constraints on physical parameters, called *intermediate specifications* (Fig. 2e). Parameters used in the intermediate specifications are called *intermediate parameters* [5, 6] and are of a rather high level. Intermediate parameters are defined as quantifiable and measurable entities referring to the physical world. Just as the parameters and the specifications, constraints established in the intermediate specifications are also called intermediate constraints. In this way, it will be possible to specify the internal characteristics of the component or eventually its links with other components.

By instance, consider the function "to be handled easily by a human–being" for a water bottle. This function can be decomposed into some constraints such as:

- Weight less than 2 kilo,
- Volume not greater than 3 litres,
- Material used must be light and so on…

In [6], we have defined an intermediate constraint by a quadruple <IP, R, Exp, W> where IP is an intermediate parameter, R is a relation among {<, >, =, ≠}, Exp is an arithmetic expression, *W* is the relative weight of constraint in comparison with the other constraints of the intermediate specifications.

Once a great number of intermediate constraints are obtained, we consider that it is sufficient to propose shapes solutions to the designer that achieved those intermediate constraints, and so that satisfied functions, from which the constraints are deduced. We suppose that this technique could be applied in a more general way.

From the product's specifications (Fig. 2a) to the intermediate constraints (Fig. 2d), actions achieved by the designer during the design process are currently manual. They might be assisted with difficulties in so far as manipulated information are mainly expressed in natural language. However, as intermediate constraints can be formalized, we can attempt to automate the intermediate constraints to form mapping in order to assist the designer. For this stage, we propose to generate the component shape of the product. Before generating the product shape that is composed of components, the complexity is reduced by generating a component after the other and next by assembling shapes of each component of the product. A component will be produced from its own intermediate parameters. Each component is a rigid, finite and homogenous solid. A set of shapes called "solutions space" (Fig. 2f) is generated for a component from primitive shapes. Naturally, several shapes could be proposed to the user: we must detect the shapes, which are best satisfying the intermediate constraints (Fig. 2g). For that, we have defined a Satisfaction Degree $SD_{ic}(s)$ of an intermediate constraint *ic* by a given shape *s* as a real number in the [0,1] interval. This number expresses the quality with which *s* satisfies the intermediate constraint *ic*. If $SD_{ic}(s)$ is near zero, the constraint is badly satisfied, if it is near one, the constraint is well satisfied. The method to compute $SD_{ic}(s)$ changes depending on whether *s* is a primitive solid or a complex object [5].

In this section, we will first detail the way to generate solutions space. Then, we will present the advantages and drawbacks of such method, and finally we will investigate how to apply it in collaborative environment.

A. Generation of the solutions space

The solutions space is composed of various topologies that are solutions (like spheres, boxes or more complex shapes...) that we call *shapes classes*. To model geometry associated with each topology, we associate each shapes class with a set of physical parameters of lower level. They are called *terminal parameters* for distinguishing them from physical parameters and are strongly related to the geometry (the radius of a sphere, the width, the length and the height of a box...). They must represent the shapes class geometry in a coherent way. For instance, a tetrahedron can be characterized indifferently by the lengths of each side of the base and the height on the one hand, or by three angles and the height on the other hand. Choosing a real value for each terminal parameter determine a solution shape.
Finally, the solutions space is generated as automatically as possible by using expert's rules alike algorithms that translate intermediate constraints into terminal constraints. As a single intermediate constraint can generate several terminal constraints, we have to compute the intersection between the all generated terminal constraints into intervals for each terminal parameter.

B. Synthesis of this methodology

First of all, one has to consider this methodology as a means of automatically generating solutions shapes that may give ideas to the designer for the final product shape. We applied such a methodology within the framework of the filling system for foundry mould design [8]. We validated the methodology by taking the specificity of the foundry (trade features) into account. We showed that this approach was applicable to this case and that we obtained quickly results better than those based on an expert's knowledge.
Moreover, the design process of this methodology consists in presenting several solutions shapes that best satisfy the intermediate constraints to the designer. Next, the design process does not stop there in so far as the designer has to modify the intermediate specifications if the resulting shapes are not good enough for him and recursively until to obtain the most promising shapes.
In fact, the previous method could be hard to use by only one person because the intermediate constraints' consistency is in charge of the designer, even if there is several thousands of constraints. In addition, the same designer ought to manipulate all the intermediate constraints in order to produce an acceptable solution shape: to do this, he ought to know the signification of all the constraints; that is rather unrealistic in most cases.
A more suitable methodology is to distinguish between intermediate constraints from a specific domain of activity and intermediate constraints from other domain of activity. As a consequence, we treat in the next section the way to apply this approach in a collaborative context with the aim of reducing productivity time but also of managing a lot of experts points of view.

C. Methodology applied in a collaborative context

As it seems to be unrealistic to design the product shape only by a designer due to the large number of design domain that intervenes, we propose to put the previous methodology in a collaborative context.
First of all, the different stages that consist in translating product's specifications (Fig. 2a) into intermediate specifications (Fig. 2e) are unchanged except the fact that there is at least one expert per design domain. Consequently, we can group the intermediate constraints based on the same design domain together (Fig. 3a): for instance, we have distinguished mechanical, electrical and thermal design.
The adaptation of the previous methodology in the

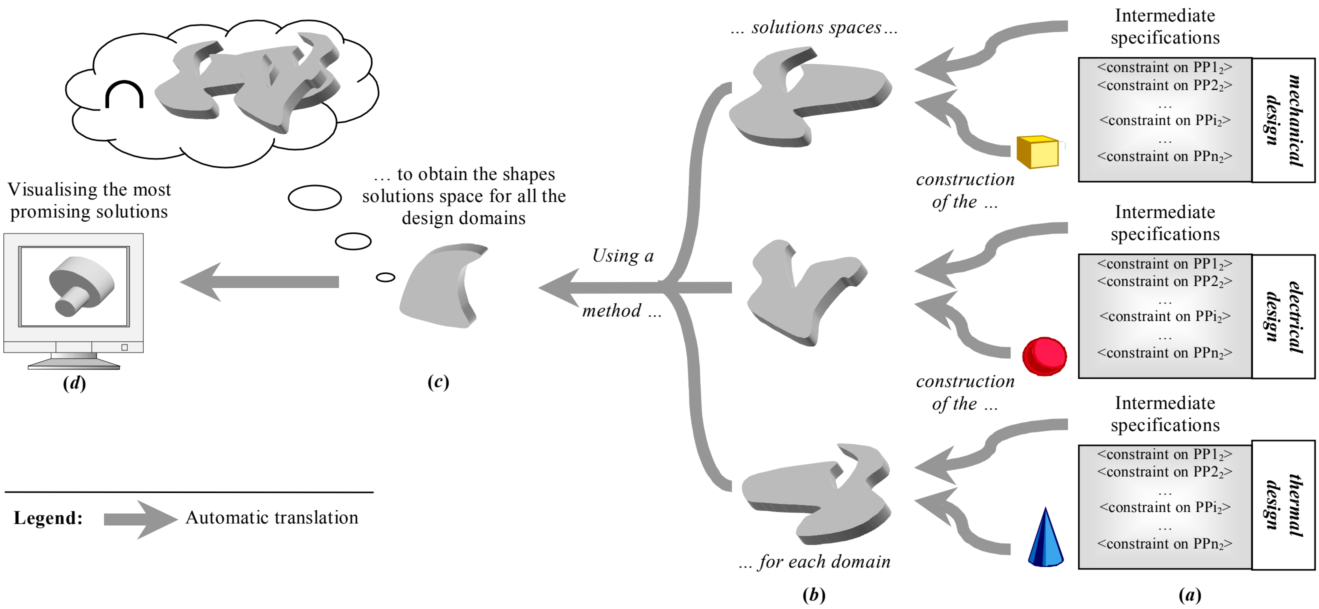

Fig. 3 New methodology in a collaborative environment

collaborative context consists in applying the method for each design domain. Consequently, each expert of one domain will generate a solutions space (Fig. 3b) from his own intermediate specifications and his own primitive shape library. Recursively, each expert can modify its solutions space by modifying its intermediate specifications as the previous method did. Once each expert is satisfied about his solutions space, one has to merge all the solutions space from the different design domain in order to find the most promising solutions that satisfied all the domains (Fig. 3c). The difficulty lies in the fact that each solutions space is composed by shape classes, and each of them is composed by terminal parameters that are totally different in theory because of their origin of different domain. As a consequence, it is unrealistic to merge all the solutions space from all the design domain because few shape class combinations could be done. Currently, techniques that come in our mind are stochastic or morphing methods as studied in [14]. Finally, once a technique will be found, one can visualize the most promising solutions using computation of the Satisfaction Degree.

III. PROPOSED MODEL FOR REPRESENTATION OF EXPERT VIEWPOINT

The analysis of current works on product modeling shows that current single fixed representations are inadequate to model the various concepts present in multidisciplinary product development situation. Consequently, the dynamic representation of multiple views of a product based on functional contexts seems to be necessary.
Depending on the view taken, certain properties and descriptions of the object become relevant. A comprehensive model of a product must be able to built depending on the particular need.
Consider the example of the mechanical part in Fig. 3d. The mechanical design model of this product is different of electrical engineer's model or thermal model. Any model should allow a dynamic evolution and must be capable of accommodating multiple concepts unambiguously and consistently so that the elements could not be duplicated. Any inconsistencies between the various models have to be discovered and corrected. This process may go through several iterations. The result is a set of models, one per consulting discipline, where, although each set represents the product using a different point of view, the comprehensive representation is consistent. There is no attempt to integrate the various sets into one set.
The basic description of a product differs from one viewer to another. Each view may represent a product with different elements and different composition hierarchies. No one model contains a full comprehensive description of the product but each model should be consistent regarding to the object being described. Different descriptions of the same elements and different subsets of these descriptions in different models exist.
Some models used for product description are shown in Fig. 4. We detail the model of intermediate specification

View point model represents the information about a product from a particular view and the relationship between the different viewpoints. Several functions are related to each viewpoint.

Purpose explains why an object does what it does and it is related to the human socio-cultural environment concept, The purpose model represents the purpose and the relationship between the purposes.

Function is what an object does. Functional model represents function and the different relationships between the functions.

Behavior is how the object does what is does. Behavior model represents the behavior, the relationships between the behaviors and the relationships with functions.

Structure is what the object is. We use the STEP standard to represent the concepts related to product data (the structure of product and its relations with other models such as shape, materials, tolerances, etc). The definition of a product in the STEP product data model is any physical object, which is produced by either natural or manufacturing processes. Any part or assembly that contributes to a product is also considered to be a product. A car is a product while its wheels and engine assemblies are considered as other products. Furthermore, each of these products can be further decomposed into smaller components or products.
The details of these models are described in [3].

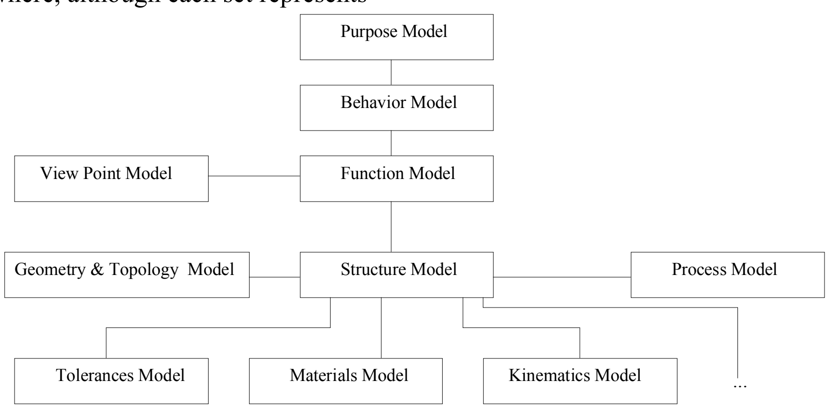

Fig. 4 Some necessary models and their relationship

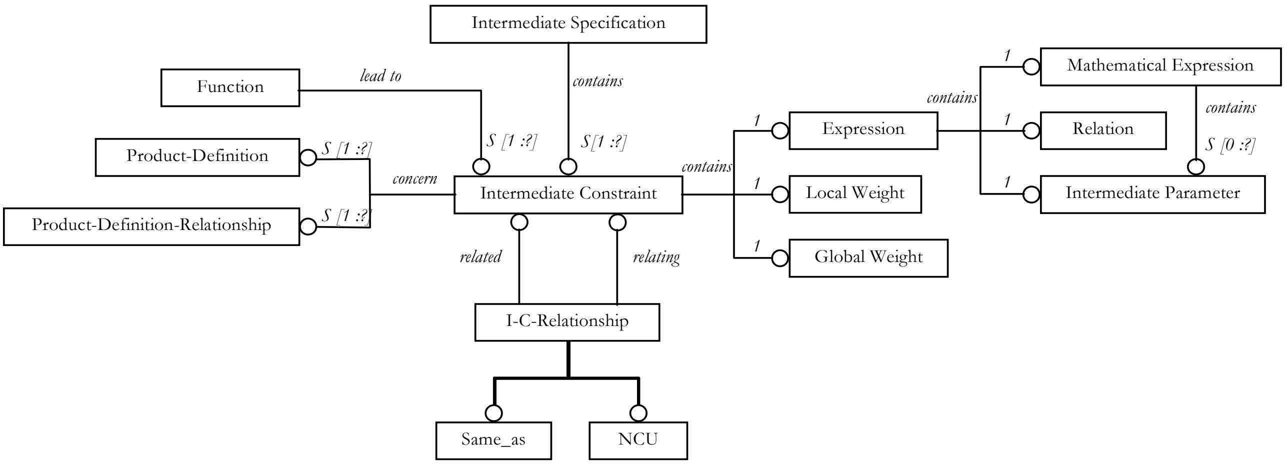

Fig. 5: Intermediate specifications model expressed in a EXPRESS– format

In this paper, due to the function form mapping technique, we have introduced the model intermediate specification. We present the detail of this model in the following.

A. The intermediate specification model

We represent the intermediate specification concepts by EXPRESS-G formalism (Fig. 5). EXPRESS-G is a graphical language developed by ISO 10303 STEP [7]. The EXPRESS-G basic notations used in Figures include *entities* (rectangles); *super-type/subtype relationships* (thick solid lines); *required attributes* (normal lines); relationship for *optional* attributes (dashed lines). Additionally, the direction of an attribute is symbolized by an open circle, where the circle represents the "many" side of a "one to many" relationship.

In Fig. 5, the i*ntermediate specifications* are the set of different *intermediate constraints* . The intermediate constraints are elaborated from the functions that characterize the product. As the functions relate to a given view point (mechanical view point, electrical view point, thermal view point and so on), the intermediate constraints also relate to this same view point. On the other hand, each intermediate constraint is defined on a component, i.e.: is related to the structure of the product (entities *Product-Definition* and *Product-Definition-Relationship*).
The intermediate constraints are expressed in the following way: first of all, an intermediate constraint described by an expression. This expression contains an intermediate parameter (which can be the volume, the weight or any engineering parameter), a relation (<, >, =, ≠, etc.) and a mathematical expression, which can possibly include other intermediate parameters. With each intermediate constraint are associated two weights: a local weight and a total weight. These weights are used to give a list of priority of the constraints that must be carried out for the expert of a precise field (*local weight*) and for all of the experts (*total weight*).
Lastly, due to the fact that each expert handles his own whole of constraints, there are relations between them (*IC-Relationship*). These relations are of two types. We define the Same_as relation, which expresses the fact that two experts handle the same constraint. Their expression and their total weight will be then identical, but their local weight might be different. We also define the NCU relation (*Next Constraint Usage*), which means that an intermediate constraint can be compose of one or more other constraints.

IV. CONCLUSION AND FUTURE WORKS

The analysis of current works on product modeling shows that current single fixed representations are inadequate to model the various concepts present in multidisciplinary product development situation. Consequently, the dynamic representation of multiple views of a product based on functional contexts seems to be necessary. In this paper, we have studied the problem of representation of experts' multiple-view in a collaborative conceptual design environment. We have considered the function to form mapping approach as a conceptual design method and we have presented the possibilities to automatically check whether a shape satisfies the specifications.
Our opinion is that whether the shape is simple (a primitive solid), or complex (combined object) this operation can not be performed without participation and collaboration of different human experts. However, it seems that the software will be more and more self-sufficient as its experience will grow, particularly for simple shapes.
Our future works concerns how to maintain the consistency of models and how to integrate the different shapes solution spaces. 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. As a consequence, when one model is manipulated, corresponding effects will be made automatically in the other. Future works will focus on the definition and formalization of these rules to improve proposed multiple-view model.

V. REFERENCES

[1] P. Ghodous, D. Vandorpe, Editors of Advances in Concurrent Engineering '2000, Technomics Publishing, USA, ISBN 1-58716-033-1.

[2] Biren Prasad, *Concurrent Engineering Fundamentals*, Vol I, II, Prentice Hall , USA, 1997.

[3] P. Ghodous, M. T. Martinez, D. Vandorpe, (2000), "**Collaborative and standard design and manufacturing model**", *International Journal of Computer Applications in Technology*.

[4] C. Dartigues, P. Ghodous, M. T. Martinez, and D. Vandorpe, (2000), "**Modélisation collaborative des activités de conception**", in *Proceedings of Conférence Internationale sur la CFAO, la Simulation et les Nouvelles Technologies de Conception et de Fabrication (MICAD'00)*, Paris, France, pp. 231-240.

[5] Y. Gardan, C. Minich, and D. Pallez, (1999a), "**On shape to specifications adequacy**", in *Proceedings of IEEE Proceedings of Information Visualisation (IV)*, D. Dornfeld (eds), Kluwer academic, London, Great Britain, July 14-17, pp. 315-320.

[6] Y. Gardan, C. Minich, D. Pallez, and E. Perrin, (1999b), "**From Functions to Shapes**", in *Proceedings of Third International conference on Engineering Design and Automation (EDA)*, Vancouver, Canada, August 1-4, pp. 287-294.

[7] ISO 10303-11, STEP Product Data Representation and Exchange (1994) `Part 11, Description Methods: The EXPRESS Language Reference Manual', International Organization for Standardization, Subcommittee 4, NIST.

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

[9] A. M. Keuneke, (1991), "**Device Representation. The significance of functional knowledge**", *IEEE Expert Intelligent Systems & Their Applications*, 6(2), pp. 22-25, April.

[10] J. McDowell, T. Lenz, J. Sticklen, and M. Hawley, (1996), "**Conceptual design for polymer composite assemblies**", *AI System Support for Conceptual Design, J. Sharpe, ed., Springer-Verlag* , pp. 377-389.

[11] T. Tomiyama, Y. Umeda, and H. Yoshikawa, (1993), "**A CAD for functional Design**", *Annals of CIRP'93*, pp. 143-146.

[12] Y. Umeda, M. Ishii, M. Yoshioka, Y. Shimomura, and T. Tomiyama, (1996), "**Supporting conceptual design based on the function-behaviour-state modeller**", *Artificial Intelligence for Engineering Design*, Analysis and Manufacturing, 10, pp. 275-288.

[13] M. Ranta, M. Mäntylä, Y. Umeda, T. Tomiyama, (1996), "**Integration of Functional and Feature-based product modelling –the IMS/GNOSIS experience**", *Computer Aided Design*, 28(5), pp. 371-381.

[14] Y. Gardan, C. Minich, D. Pallez, and E. Perrin, (2000), "**Towards a specifications-to-shape translation tool**", in *Proceedings of Third International Symposium on Tools and Methods of Competitive Engineering (TMCE)*, Delft, The Netherlands, April 18-21.