How to Represent “Intelligent” Components in a Product Model

A Practical Example

L. Susca[1], F. Mandorli[2] & C. Rizzi[1]
[1] *Dipartimento di Ingegneria Industriale, Univerità di Parma*
[2] *Dipartimento di Ingegneria Meccanica, Univerità di Ancona*

Abstract: This paper presents a multi-level approach to define a product model. It is based on the concept of what we call “Intelligent” Component. In order to be able to manage contextually the different types of knowledge involved during the design process, the multi-level model reflects the different steps of the process itself. To describe the approach an applicative example related to shaft design has been implemented. We first illustrate how to define an “Intelligent” Component for shaft design, and, then, how to extend a single-part approach to a library of mechanical "intelligent" components that allow developing complex models. It permits to show how a multi-level product model is able to capture and represent the design process from the preliminary to the detail stage, formalising all the information concerning the behaviour of the model within different application contexts.

Key words: Product model, intelligent component, multi-level architecture

1. INTRODUCTION

The reduction of time-to-market and the production of high quality and low cost products are the tight challenge that manufacturing industries have to face in order to cope with the ever-increasing worldwide competition. To reduce time-to-market, the right design must be identified as soon as possible; ideally, the solution can be summarised as follows: *right design the first time*.

As stated by a wide literature (see, for example, Woodson 1966, Yoshikawa 1981, Shigley 1983, Middenford 1996, Tomiyama 1987, and

SME 1993), the engineering design task is an iterative decision-making process. Design itself is continuously evaluated and changed in order to satisfy all the conditions that are imposed not only by design specifications, but also by all the physical, technological, and marketing constraints related to the manufacturing process.

It is widely accepted that the design activity can be subdivided into four phases: requirements definition, conceptual design, embodiment design, and detail design (Shah 1995, Bozza 1998). The objective of the first phase is to set the performances and the overall constraints the product must satisfy (design specification). This phase is followed by the critical task of the conceptual design. During this stage, the functional structure of the product is defined in order to meet the design specification identified before: different solution principles are analysed, selected and combined to define alternative conceptual solutions (Tomiyama 1993). The reasoning performed during this activity is usually done at a high level of abstraction and on the basis of the designer's experience. The objective of the embodiment phase is to translate the conceptual solutions into layouts and rough shapes. Once the best design solution has been identified and refined, the detailed design takes place: final shapes with dimensions and tolerances are set as well as materials and manufacturing process. The result of the detailed design phase is the generation of all the documentation required for the production process.

However, the evolution of CAD systems has followed a *shape-oriented* philosophy: from 2D drafting systems to sophisticated 3D modelling systems, the focus has always been the development of technologies which provide the user with powerful tools able to represent the geometrical aspects of a product.

Storage and management of the choices that the designer has made during the reasoning from the design specification to the design solution, is far beyond the capability of these systems.

Conversely, identifying the best design solution in the early stages imposes to move from *shape-oriented* systems to *knowledge-intensive* design systems, where shape is just one of the types of knowledge the system must be able to manage. We think that the identification of appropriate structures and frameworks for the definition, integration and management of different types of knowledge at different level of abstraction is the basis for the development of Knowledge Intensive CAD systems.

In this paper we present a multi-level approach for the definition of a product model. It is based on the concept of "Intelligent" Component, and reflects the different steps of the design process, from the preliminary to the detailed one, in order to be able to manage contextually different types of knowledge. For a better comprehension of the approach, a multi-level model related to shaft design has been implemented and described in the following chapters.

2. ADOPTED APPROACH

By using traditional parametric CAD systems, it is possible to model components whose dimensions are computed on the basis of formulas/rules linked to geometrical parameters. It is then possible to define spatial relationships (mating between planar surfaces, co-axiality among cylindrical surfaces, etc.) among the modeled components in order to define the assembly model.

This modeling process forces the user to follow a bottom-up approach that has the following drawback: it is not possible to support the definition of the appropriate dimensioning rules for the components because the system has not the knowledge about the overall structure of the product, not yet defined. Moreover, if the user wants to use the same component within a different product model, he is forced to re-define the dimensioning rules in order to meet the requirements of the new product.

On the contrary, as reported in chapter 1, the design process is an activity that couples the bottom-up approach with a top-down analysis. The overall structure of the product is top-down analyzed in order to meet the design requirements, and it is, step by step, refined down to the definition of the shape and the technological properties of the components to be bottom-up assembled.

Following this approach, the selection and dimensioning of a part to be inserted into a mechanical system, come from the application of generic functional and technological principles applied to the specific component and its application context. This can include considerations about the overall structure of the product, materials and stress conditions, functionality related to the contact surfaces, bounding volumes, technological constraints, etc.. For example, while designing a reduction gear, the designer selects the appropriate bearings from a catalogue on the basis of the specific application context. The bearings listed in the catalogue represent the application of engineering principles that associate bearing types and dimensions to related working conditions. The analysis of the product allows the designer to extract the parameters to be used as input for the selection.

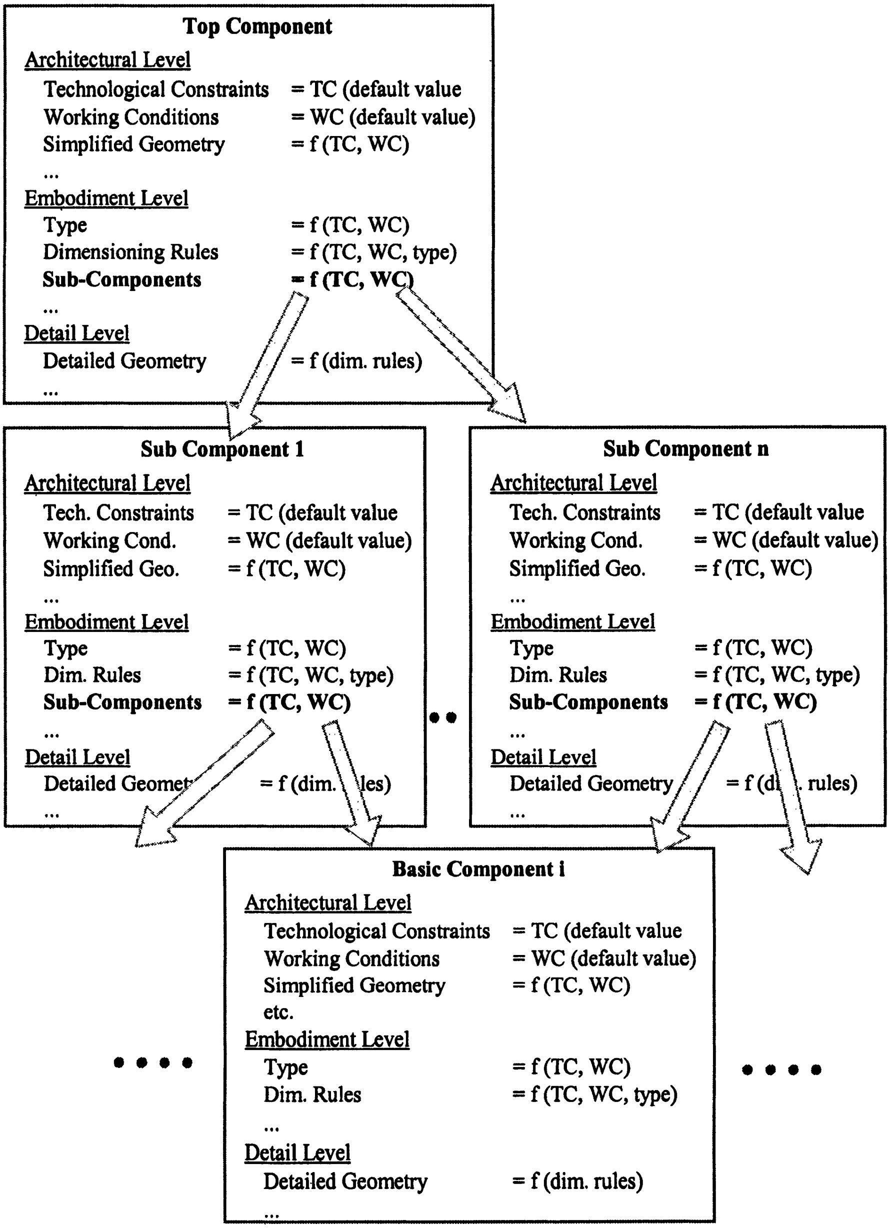


Figure1. Product model structure

Our objective has been to define a multi-level model of a product in order to capture and represent the complexity of the design task. It reflects the steps of the design process, from preliminary to the detail one.

The multi-level model is a hierarchical structure able to represent the product model at different levels of abstraction (different steps of the design process) used by the designer during his/her reasoning. Each abstraction level encapsulates and formalizes the knowledge related to the corresponding step, and inherits constraints and results coming from the reasoning performed at the upper levels of abstraction.

To define a multi-level model we introduce what we called "Intelligent" Component (IC), which can be viewed as an extension of semantic feature concept. Features have been widely used to approach the problem of knowledge representation in design support systems (Shah 1994, Shah 1995). They can be used to represent part of a component shape with an associated functional and/or technological meaning (semantic) within a specific application domain (Mandorli 1997, Bidarra 1999). In our approach, we define feature components as semantic objects having appropriate methods which allow the integration of the component within the product model, and provide control to its correct behavior.

An IC is an object that includes different types of knowledge corresponding to different levels of abstraction. The types of knowledge are related to different aspects of the product, such as shape (e.g., dimensioning rules), functional meaning, material, etc..

Figure 1 shows a generic product model with different multi-level components and sub-components.

At the first level of abstraction, called architectural level, the component stores the representation of its overall structure in terms of the simplified geometry of its main sub-assemblies and components (reference planes, axis, bounding boxes, interface between components, etc.) as well as constraints (both geometric and not geometric) and relationships among them. Lower levels and related knowledge are organized according to the sub-sequent steps of the specific component design process: embodiment level, ..., detail level. The component is also provided with methods to scan the product model structure (where the component will be inserted), in order to retrieve the information that is required as parameter for the rules that will drive the component behavior. These methods will benefit from the multi level organization of the knowledge in the sense that, depending on the type of knowledge the method is looking for, it will know at which level of abstraction it should find it. When the user needs to define the product at a lower level of abstraction, he can benefit from the knowledge stored at the upper level that will drive the behavior of the components represented at the lower level.

Therefore, we can consider a product model as made by "Intelligent" Components that have the capacity to encapsulate the knowledge required to represent functional aspects as well as behavior in respect with the specific application domain and level of abstraction.

This leads to the definition of a library of "Intelligent" Components, and set of rules that define how the components should behave or configure themselves depending on the application context.

3. PRACTICAL EXAMPLE

The next paragraphs describe an application based on the approach described; in particular, the following problems will be faced:

- Implementation of an IC for shaft design (§ 3.1);
- Extension of the single-part approach to a library of mechanical IC that allow developing complex models (§ 3.2).

3.1 Example of an intelligent component

As a reference example to validate our approach, we have considered the problem of shaft design. The development of the model has followed two main steps:

1. Analysis of the design process;
2. Formalisation of the design knowledge into the model structure.

The knowledge was captured and formalised, using an appropriate software tool (Selling Point, www.oracle.com/applications/sellingpoint), which allows representing the design stages into a hierarchical structure.

3.1.1 The shaft design

The shaft is a mechanical element, which allows to support and link rolling machine members. The possible kinds of constraint, load, cross section and axis, identify different shaft classes (see table 1), which require appropriate design rules. Within this example, we consider only shafts with circular cross section and straight-line axis, even if it could be possible to extend the analysis to the other classes, by introducing all the necessary rules and knowledge.

Table 1 - Different kinds of shaft

Parameter	Possible Shaft Type	
Axis	straight-line shaft	Crankshaft
Cross Section	circular shaft	section bar (for heavy torque)
Load	(no torque) spindle	(torque) shaft

Main steps of the design task of shafts can be summarised as follows:

Context definition: during this stage, the designer defines general requirements, as constraints, loads, and overall dimensions.

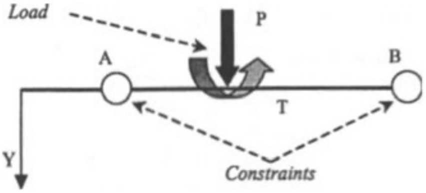


Figure 2. Architecture of the shaft with functional constraints

Only simple geometry is required in order to represent the architecture of the system (see figure 2).

1. *Stress/strain analysis*: after establishing the structure of the shaft with external supports and loads, the designer has to perform the stress/strain analysis, by applying the construction theory, in order to evaluate the minimum value of the shaft diameter. In this phase s/he has also to identify the material to be used and the fatigue limit, which determines the life of the part.
2. *Shaft shape definition (no 1):* the new results achieved allow the designer to define an approximated shape of the shaft, which depends on the position of supports, loads, and on the minimum diameter computed.
3. *Shaft shape definition (no 2)*: to verify and complete the first shaft shape, it is necessary to evaluate other important functional parameters, as:
   - Vibration critical speed;
   - Coupling parameters (for example bearings internal diameter, gears dimensions and tang / key size).
4. *Manufacturing process considerations*: information dealing with the manufacturing process permits to identify the final shape of the shaft. The production volume, the cost analysis, and the weight/quality requirements represent the main constraints for the designer, who has to decide if a cheap or high quality/expansive shaft should be produced.

3.1.2 The model of the shaft

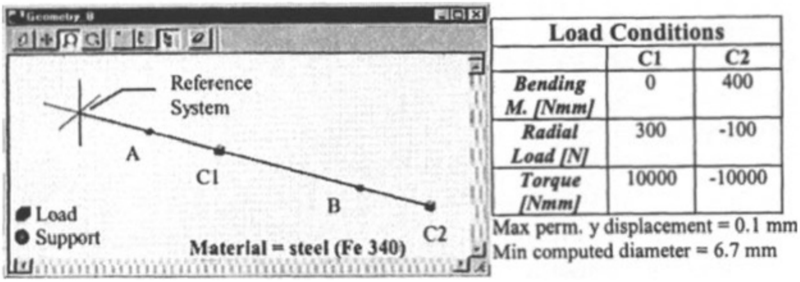
The design process described has then been organised into the shaft model. Different levels, which are represented by the child-nodes of the whole shaft model, provide the designer with the knowledge used to perform the various steps of the process. The logical sequence of decisions and their relationships have been implemented within the model by connecting the results and design variables managed at the various levels. The user interacts with the model by editing the design parameters and by modifying the decisions previously taken. The model re-executes the design process from the first level or stage to the last one and provides the designer with the new solution computed. In this way all the computations and repetitive tasks of the process are performed automatically, while the designer can spend his/her time in order to improve the solution. The different levels of the shaft model, which correspond to the stages of the design process, can be summarised as follows:

level 1 (steps 1-3): the end user sets the general requirements (overall dimensions, loads, supports, permissible camber, material, ...); the model applies the construction theory to the shaft architecture (figure 3) and performs the stress/strain analysis by computing a fixed number of cross-sections. The main results calculated are:

- Chart of bending moment, shear, and torque (figure 4);
- Camber chart (figure 4);
- Minimum diameter that assures the shaft bending within permissible camber (figure 5);

rough shaft shape, which is composed by several frustums of cone whose diameters come from the stress/strain analysis of the shaft cross-sections (figure 5).

level 2 (step 4): The model converts the rough shaft shape into an even one (figure 6) by comparing the computed diameters of the different cross-sections to the available standard diameters. The model supplies a refined shaft shape which consists of few solid primitives (cylinders and cones); the designer can interact with the model setting the approximation rate that drives the model into the shape refinement process.



Material = steel (Fe 340)

Load Conditions		
	C1	C2
Bending M. [Nmm]	0	400
Radial Load [N]	300	-100
Torque [Nmm]	10000	-10000

Max perm. y displacement = 0.1 mm
Min computed diameter = 6.7 mm

Figure 3. Shape architecture with load conditions

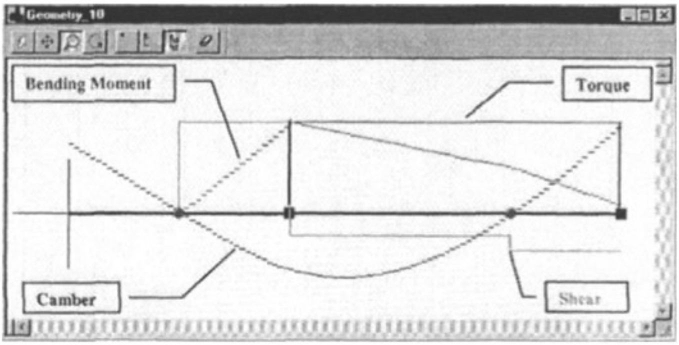


Figure 4. Charts with shaft architecture

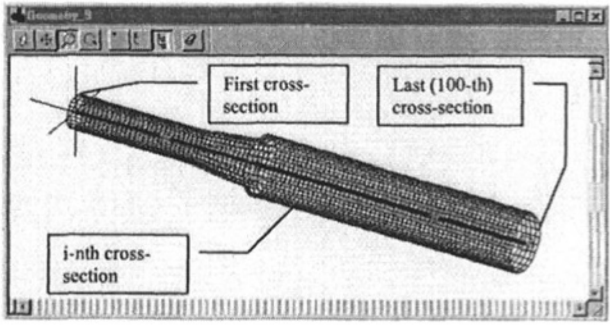


Figure 5. Rough shaft shape with architecture

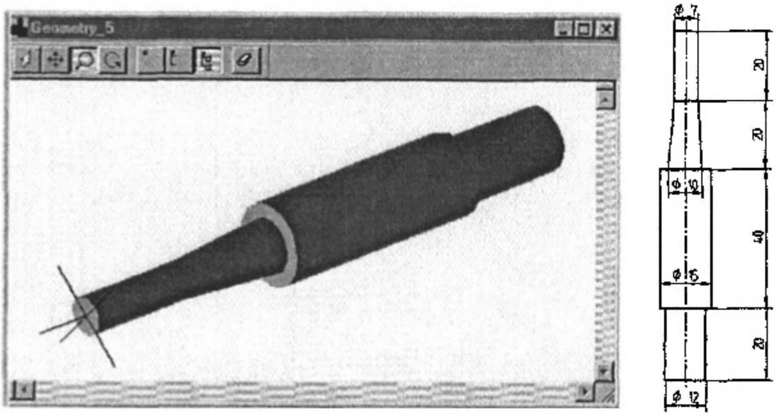


Figure 6. Refined shaft shape

level 3 (step 5): The refined shape model represents the smallest shaft that satisfies all the functional and structural requirements. Other considerations, dealing with the manufacturing process, allow deciding if this model meets also the cost requirements. This problem has led to the introduction within the model of a new level related to a further optimisation of the shaft. The end-user sets the value of a particular cost parameter and the model decides which of the possible solutions based on the refined model should be adopted (figure 7).

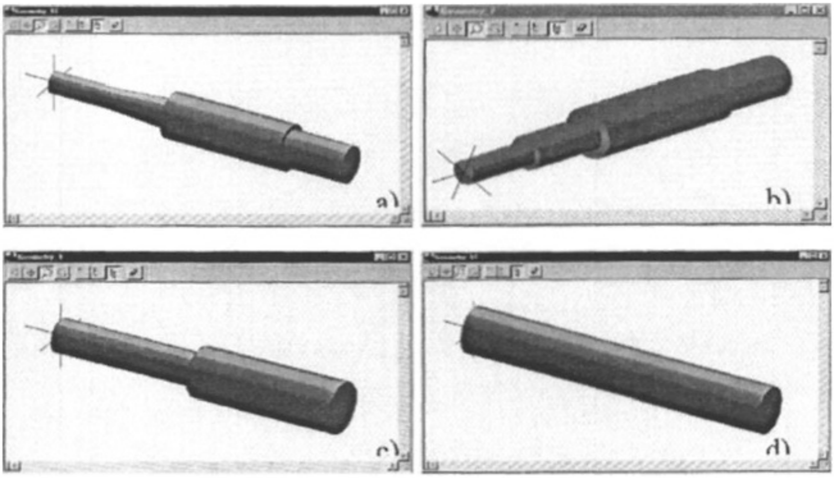


Figure 7. Different configurations of the shaft: a) expansive and light shaft, b) medium value of cost parameter (only cylinders), c) low/medium value of cost parameter, d) cheapest shaft

The shaft created by the user during the different design stages should be provided with the assembly features (for example shoulders, grooves, slots, and key-ways) required by all the elements, which will couple to the shaft itself.

In the context of the example presented, we developed the assembly model of the shaft with the bearings (see figure 8). The model of the bearing, as the shaft one, includes all the knowledge needed to perform the design process that, in this case, consists of various rules needed to select the right component within the supplier handbook (see next paragraph for the extension of the shaft design approach to other mechanical systems).

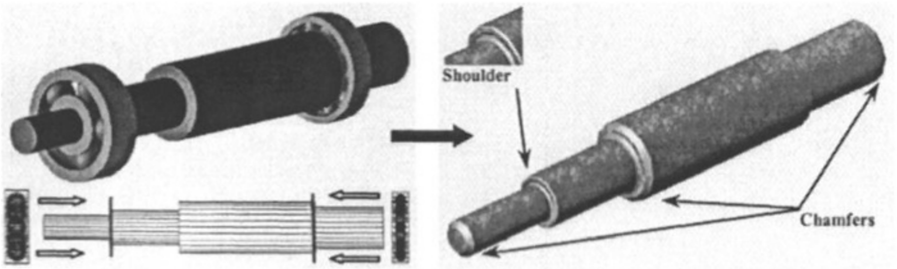


Figure 8. Shaft with bearings and detailed model (Solid Edge)

In order to complete the shaft and introduce further details (figure 8) a 3D CAD tool (Solid Edge, www.solidedge.com) has been integrated to the multi-level system.

The detailed shaft model can be finally verified by using an appropriate simulation tool for FEM/FEA. As a result the designer can compare the information provided by the multi-level model (for example shaft stress and strain) to the data coming from the simulation test (table 2). Figure 9 shows the model related to the study-case of this paper as it appears after the test performed with a commercial simulation tool.

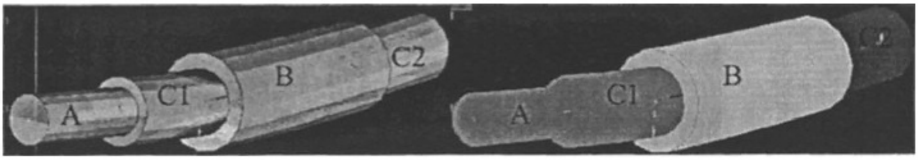


Figure 9. Finite Element Model of the shaft: a) stress and y displacement, b).torsion

Table 2 - Shaft Data Comparison.

Multi-Level Model		Finite Element Model	
Max y displacement [mm]	0.018	Max y displacement [mm]	0.009
Max y displacement [mm] (cons. shaft diam. = 7 mm)	0.095	Max y displacement [mm] (cons. shaft diam. = 7 mm)	0.096
		Max torsional stress [Mpa] (see fig. 9/b)	29.7
		Max stretch [Mpa] (see fig. 9/a)	48.8
Fatigue Limit [MPa] (safety factor = 2)	54	Max total stress [Mpa] $(\sigma^2 + 3\tau^2)$ - (in C1, fig. 9)	$(40^2 + 3*19^2)$ = 51.79

3.2 Extension to a mechanical system

The approach adopted for the development of the shaft model can be extended to other machine members in order to build up a library of Intelligent Components. The end-user can develop complex models combining the available members included within the library. In order to verify this opportunity a model of a reduction gear has been implemented and tested.

Shafts, gears and bearings are the main elements, which constitute the system. Different assembly constraints allow defining the general structure, which represents the contest of each component. For example, the design configuration of the bearings depends on the shaft diameter and on the support pressure, which can be computed after defining the functional requirements of the reduction gear. The introduction of new general parameters (for example: power, gear ratio, and angular velocity) allows the end-user to interact with the model from a general point of view as if it could be considered a new independent component. The single parts update themselves automatically thanks to the assembly relations previously imposed.

The model of the reduction gear represents a new Intelligent Component which includes the knowledge needed to perform the whole design process; for this reason it can be re-used for the development of other mechanical systems.

CONCLUSIONS

In this paper, we have presented a multi-level approach to define a product model. It is a model that reflects the different steps of the product design process and permits to manage contextually the different types of knowledge involved during the process. Motivations of our work arise from the fact that most of the commercial systems cannot represent the multiple choices the designer performs during her/his reasoning from the design specification to the design solution. The evolution trend of design support systems focus on the capability to represent product design process instead of only the product itself:

- Traditional CAD systems: the designer generates a pure geometric model, describing the object shape, that is the final result of a complex reasoning, not stored in the model.
- Parametric/feature-based systems: the geometry is enriched with information of different nature, trying to capture the designer's intent and knowledge. The designer describes the object shape by means of basic entities (parametric features) which are associated to a meaning (functional, technological, etc.) is stored in the model. What is missing is *why* the designer has used those features or parameters
- Knowledge-based and configuration systems: the designer can include within model different types of knowledge capturing and formalizing most of the *decision process,* s/he followed to design such a product. However, the product evolution through the different steps characterizing the design process is completely loss. Given a fixed set of input data (dimensions, material, cost, etc.) the product model obtained is always the same as well as the decision process. This is not always

true; in fact, even if the designer considers the same initial constraints and the same product architecture, s/he can perform different choices at the intermediate steps that lead to an evolution of the product itself.

The previous chapters have shown how a multi-level product model is able to capture and represent different types of knowledge characterizing the design process. This allows formalizing all the information concerning the behavior of the model within different application contexts. We talk about "Intelligent" Component in relation with the capability of multi-level knowledge-based models to fit different situations and to provide the right configuration once the designer has defined the general requirements. The models based on this methodology can be used as constitutive parts of more complex systems that inherit from their components the knowledge and the levels of abstraction needed to behave "intelligently".

Adopting a multi-level model, a product is the result of a set of choices done at the different levels of abstraction, recovering, in this way, the component history/evolution through the design process phases: from sketch to detail drawing.

REFERENCES

Bidarra, R. (1999). *Validity Maintenance in Semantic Feature Modeling,* Thesis Technische Universiteit Delft, Universal Press, Science Publishers, Veenendaal, The Netherlands, ISBN 90-9012599-X.

Bozza, I., Cugini, U., Ruozi, D., and Petrotta, G. (1998). Modeling and Simulating Improvements of the Ferrari Auto Steering Gear Development Process, In *Proceedings of 31st ISATA International Symposium on AutomotiveTechnology & Automation,* Advanced Manufacturing/Innovation Management, ISATA, Croydon, UK, pp. 519-525.

Yoshikawa, H. (1981). General Design Theory and a CAD System, In *Man-Machine Communication in CAD/CAM - Proceedings of the IFIP Working Group 5.2-5.3 Working Conference,* T. Sata and E. A. Warman (eds.), pp. 35-53.

Mandorli, F., Cugini, U., Otto, H.E., Kimura, F., (1997). Modeling with self Validation Features, In *Proceedings of ACM/IEEE Symposium on Solid Modeling and Applications '97,* pp. 88-96.

Middendorf, W.H. (1996). *Design of Devices and Systems,* Marcel Dekker Inc., New York.

Shah, J.J., and Mantyla, M. (1995). *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, Applications,* John Wiley & Sons, New York, NY.

Shah, J.J., Mäntylä, M., and Nau, D. (1994). *Introduction to Feature Based Manufacturing, Advances in feature based manufacturing,* Elsevier Science B.V.

Shigley, J,. and Mitchell, L. (1983). *Mechanical Engineering Design,* McGraw-Hill, New York.

SME, Society of Manufacturing Engineers, (1993). *Tool and Manufacturing Engineers Handbook, V.5, Manufacturing Management, Ch 13, Design for Manufacture,* R.F. Veilleux and L.W. Petro (eds.) Fourth edition.

Tomiyama, T., Kiriyama, T., Yoshikawa, H. (1993). Conceptual Design of Mechanism: A qualitative Physics Approach, In *Concurrent Engineering: Automation, Tools and Techniques*, A. Kusiak (eds.), John Wiley & Sons.

Tomiyama, T,. and Yoshikawa, H. (1987). Extended General Design Theory, In H. Yoshikawa and E.A. Warman (eds.): *Design Theory for CAD, Proceedings of the IFIP Working Group 5.2 Working Conference,* pp. 95-124.

Woodson, T.T., (1966). *Introduction to Engineering Design*, McGraw-Hill Book Co., New York.