Proceedings of the 2nd WDK Workshop
on Product Structuring
June 3-4, 1996
Delft University of Technology
Delft, The Netherlands

Issues in Product Structuring

Marcel Tichem and Ton Storm
Laboratory for Production Engineering
Delft University of Technology
Landbergstraat 3
NL-2628 CE Delft
The Netherlands
tel: +31 15 2781603/2786580 fax: +31 15 2783910
e-mail: (m.tichem/t.storm)@wbmt.tudelft.nl

Abstract

This paper provides an introduction to the field of product structuring and summarizes the results of the 2nd WDK Workshop on Product Structuring. It is argued that two main interpretations of product structure can be distinguished, based on the interest of people working in the field. The first interpretation, referred to as the design interpretation of product structure, focuses on the creation of product structures. Work done in this interpretation is directed towards defining product models, defining models of the design process and developing designer support. The second interpretation, referred to as the design data management interpretation of product structure, focuses on the management of design data rather than on the creation of this data. Work done in this interpretation is directed towards defining data frameworks and developing support for data management. Both interpretations are discussed using material presented during the workshop.

1 Introduction

This paper summarizes the findings of the 2nd WDK Workshop on Product Structuring. The workshop was the second one on the topic product structuring. The first one took place in 1995 [1]. Originally, the paper served as opening presentation of the workshop, and intended to provide an introduction to the field of product structuring. This second version of the paper was created after the workshop took place, having all contributions at the author's disposal. It intends to provide a global view on the results of the workshop.

The paper provides a description of the field of product structuring. When studying work done in the field of product structuring, two interpretations of the term product structure can be distinguished. The first interpretation focuses on the creation of design data, in particular product structures. The second interpretation focuses on management of design data.

In this paper, both interpretations are introduced and supported by material mainly presented during this workshop. Section 2 introduces the classification of the field of product structuring in two interpretations. In the sections 3 and 4 both interpretations are treated in more detail respectively. Section 5 closes the paper with concluding remarks.

The paper does not intend to provide full summaries of the contributions. Rather, an attempt is made to provide an introduction to the general approach and results of the contributions, and to indicate the relationships between contributions. Because of its summarizing nature, the paper contains mostly references to the proceedings [2]. References to papers included in the proceedings are made by quoting the name of the first author of the paper. These references are not included in the bibliography at the end of the paper. References to literature not included in the proceedings are made using a number between square brackets (*e.g.* [1]).

2 The two interpretations of product structure

When discussing the topic of product structure and product structuring, two main interpretations of these terms appear to exist, see figure 1.

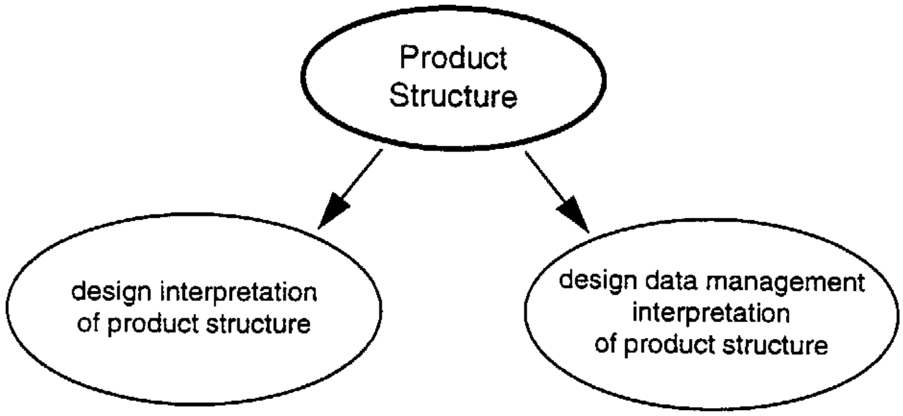

Figure 1: Two interpretations of product structure and product structuring

The first interpretation starts by defining structure as the elements and relations a system consists of. A system *has* a structure, it is a characteristic of the system. Product structuring then is the design activity of creating product structures (or more correct: a product which has structures). Product structuring must be regarded as an integral part of the design process.

In the second interpretation, a product structure is mainly considered as the carrier of design data. The product structure is in this interpretation the framework, defining the data to be captured. Design data is the collective term for all product related data (of which the "pure" product defining data is only a part) and design process related data. The difference with the first interpretation is that work done in this field focuses on the capturing and management of design data, rather than on the creation of the data. The second interpretation is closely related to defining the users of design data and their data needs.

The two interpretations should not be separated as two entirely different fields. The difference between the two interpretations is rather a difference in interest of the work done in the field.

In this paper, the first interpretation is referred to as the *the design interpretation* of product structure, the second interpretation is referred to as *the design data management interpretation* of product structure. In the next sections, both interpretations are elaborated.

3 The design interpretation of product structure

In the first interpretation, structure is defined as the elements and relations a system consists of, structuring as design activity is concerned with generating the elements and relations. This is done based on both functional and life-cycle related considerations. The "language" a designer uses to model his design consists of a variety of elements and relations, *e.g.* functions, solution principles, functional surfaces, parts, spatial relations, hierarchical relations, etc. Examples of the structures that are created during design are a structure of functions and a hierarchy of physical elements (parts). It must therefore be emphasized that the designer creates during design *a number of structures.* In addition, other disciplines in the company, like sales or assembly, also need their own view(s).

From this discussion, a number of fields of interest can be indentified in the design interpretation of product structure, see figure 2:

Defining product models. This issue is concerned with defining the language of a designer, *i.e.* the elements and relations the designer uses when modelling his design.

Modelling the process of structuring. This issue is concerned with defining models of the design process, in particular of the creation of product structures. These models should show how a structure grows, based on functional and life-cycle considerations.

Developing support for the design process. This issue points at designer support in the entire design process, in particular in structuring. Support can amongst others aim at creating the product models ("-CAD functionality") and at decision making.

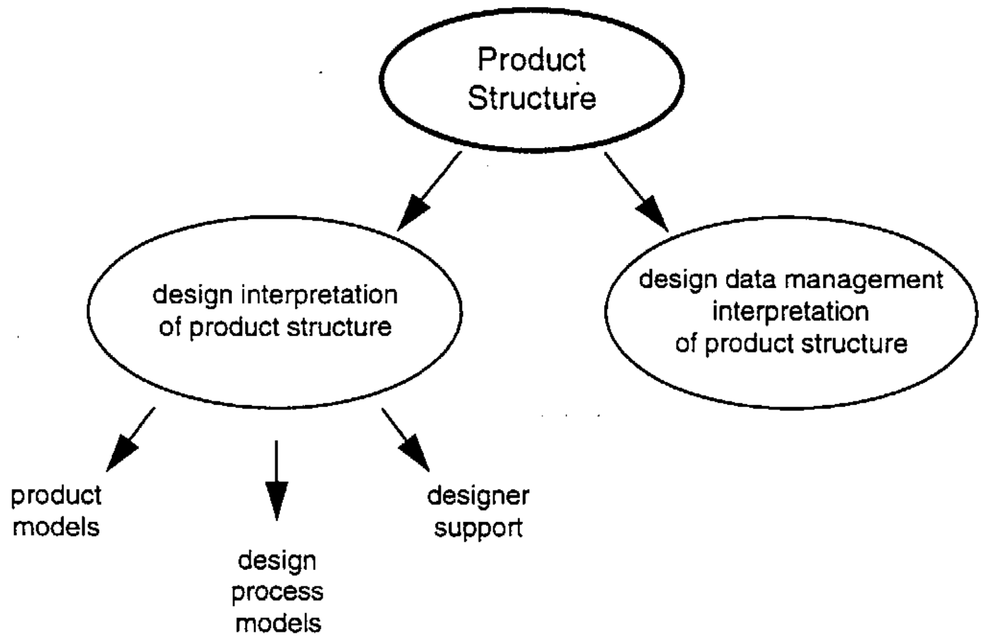

Figure 2: Issues in the first interpretation of product structure.

3.1 Defining product structure models

A product structure can be defined as a viewpoint dependent description of the build-up of a product, showing the elements the product consists of and their relations. Depending on the type of elements and type of relations regarded, a number of different structures for one product may be identified.

Andreasen summarizes the structures that can be distinguished as shown in figure 3. Andreasen distinguishes four domains used by designers to model products: the process domain, the function domain, the organ domain and the component domain. Each domain must be regarded as a unique description type of the product. The process domain describes the product as a structure of processes, the function domain describes the product as a structure of functions etc. Apparently, in each domain structures can be defined.

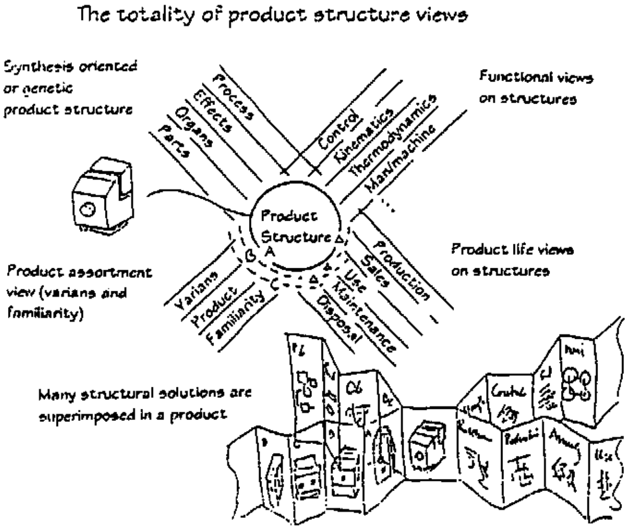

Figure 3: The totality of product structure views (Andreasen).

In addition to this dimension, Andreasen shows that for each product, functional views can be defined, corresponding to different tasks a product must perform and to different disciplines involved, *e.g.* the kinematic system, the control system etc. Again, each view has its own structures. The two other viewpoints distinguished by Andreasen show the product life view and the product assortment view on structures. The product life view shows the structures as regarded by the various life-cycle phases, *e.g.* production, distribution or service. The product assortment view is related to creating product families rather than single products.

A well-known structure type is the hierarchy, *e.g.* the functional hierarchy or the component hierarchy. A second structure type often discussed is the architecture Erens defines an architecture as the composition of a product from a number of component products. A product architecture describes the components, together with their interfaces and operation. Erens distinguishes three domains, the functional, the technology and the physical domain. In each domain, architectures can be defined on each level in the hierarchy. Figure 4 illustrates product hierarchies and architectures for the functional and the physical domain. The functional architecture shows the input-output relations between functions, *i.e.* the sequence of transformations. The component architecture shows the contacts that exist between parts, where for each contact attributes may be defined like type of contact, assembly information, etc. Lanner defines product architecture as follows: "a product architecture is the way in which the functional elements of a product are arranged into physical units, and the way in which these units interact". In this definition of architecture, the mapping between different domains is included. Erens considers it a design task to create coherence between architectures from different domains.

Hansen focuses on defining machine system relations. The basis for the work is the Theory of Domains, which states that a product may be described in four different domains (the process, function, organ and component domain), and that for each domain structures may be defined (see figure 3). Hansen investigates the relations in two domains, the component and organ domain, studying a tumble dryer. Based on this

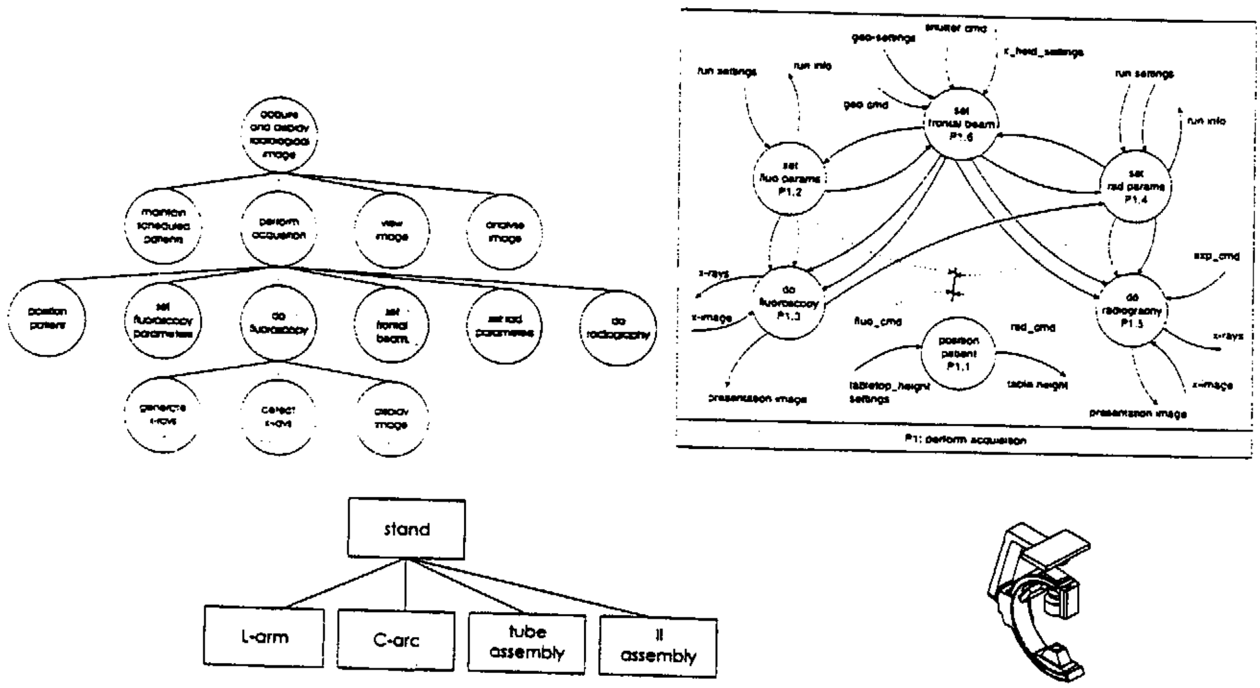

Figure 4: Functional and physical product hierarchies and product architectures (Erens).

case study, Hansen identifies various types of relations, *e.g.* the control relation, which changes the state of a machine system, or a performance relation which aims at a required performance. An example of the latter is the relation between the motor and the drum. The motor has to drive the drum with the required speed (performance relation). To solve this relation, a gear may be added to adjust the rotational speed of the motor.

3.2 Models of structuring

This subsection considers the design activity structuring. During the first workshop models of function oriented decomposition [3] and of configuration design [4] were presented. During this workshop, considerable attention was paid to developing product families, see Erens, Andreasen, Hildre, Erixon, and Lanner. Their contributions will be discussed in this and the following subsection.

A distinction must be made between variance and familiarity. Variance describes the size of the product family. Familiarity, also referred to as commonality, points at standardization. The goal of diversity management through design is to get the highest variance by the highest familiarity. These demands are conflicting, and it is an important and difficult task for designers to find a proper balance.

Erens argues that in designing a product family having a well thought architecture is of foremost importance. They observe that, increasingly product architectures are defined before the development of the product is started. In a well-developed architecture, the stable and changeable aspects of design are separated. Among the benefits of having such an architecture are increased stability, easy communication, better control of diversity demands, and improved possibilities for reuse and upgrading.

Andreasen describes structuring, taking care of functional and life-cycle considerations, including diversity demands, as a process of superimposing structuring principles. Each view[1] applies its own structuring principles. In creating a product which is structured from all viewpoints, these structuring principles have to be superimposed.

The design of a product (family) cannot be regarded as an isolated activity. A product (family) and its life-cycle processes must be developed in mutual coherence. Bikker and Erixon emphasized the need for integrated product and process design, in order to get a simpler process, better logistics, reduced control load etc. The higher levels of structuring, *i.e.* the definition of subassemblies and modules, have a high impact on the performance in production. The relevance of structuring for reducing the complexity of the assembly process is well-known. Bikker argues that for the parts manufacturing process, structuring may reduce the complexity of the process as well. In particular the control load may considerably be reduced.

3.3 Support in structuring

Support in structuring focuses in particular on the decision making in structuring. Support varies from providing principles, providing structured procedures or providing structured tools.

Hildre presents principles for designing product families. He distinguishes F- and f-familiarity; F-familiarity corresponds to variance, the demand for diversity from the market, f-familiarity corresponds to commonality. In treating F-familiarity, Hildre describes principles starting with investigating the widest possible customer demands, which is translated in a number of steps into segments. The segments are the basis for defining product families. Principles for managing f-familiarity aim for instance at increasing reuse, add customer specific modules as late as possible in assembly, and create modular designs.

Lanner presents an approach to designing product architectures, the Product Architecture Design (PAD) methodology. In Lanner's definition, a product architecture defines the mapping between functional and physical units of a product. First, an organ structure must be created, *i.e.* a structure of function carriers. In the so-called Lanner-matrix, the relationships between the various organs are quantified using technical and economic criteria; the strength of the relationship between organs determines which organs should be combined into a physical unit. In the Lanner-matrix, defining modules can be depicted as shuffling rows and columns, see figure 5.

	O1	O2	O3	O4	O5	O6
Organ 1 O1	1 3 3 3	+2 0 0 0	+2 -2 0 -2	+2 0 0 -1	+2 0 +1 0	
Organ 2 O2	+2 0 0 0	9 9 3 3 3 9 9	+2 0 0 0			
Organ 3 O3	+2 -2 0 -2	+2 0 0 0	3 9 1 9 3 9 9	+1 0 0 0		
Organ 4 O4	+2 0 0 -1		+1 0 0 0	3 1 3 9 1 3 3	+2 0 0 0	+2 0 0 0
Organ 5 O5	+2 0 0 0			+2 0 0 0	9 3 1 1	
Organ 6 O6				+2 0 0 0		9 9 9 3

Figure 5: The Lanner-matrix, part of the Product Architecture Design (PAD) methodology (Lanner).

Erixon presents a method for module definition, called Modular Function Deployment (MFD). Modules are

[1] See figure 3 for the views Andreasen distinguishes.

created for various reasons. Erixon defines *module drivers* as the criteria behind modularization, which originate in the entire product life-cycle. Examples of module drivers are the demand for reuse of former designs (designing generations of products), expected technology pushes, developing product families, replacement of modules in service, etc. Modular Function Deployment starts by translating the voice of the customer into technical requirements using QFD, see figure 6. Next, functional decomposition and search for solutions must be carried out. Using a concept selection method, the best technical solutions are selected. In the next step, a Module Indication Matrix is created, which is used to indicate for each technical solution if there is a need to create modules, and how strong this need is. Finally, proposals for modular designs are evaluated, and for each module a DFX analysis is to be carried out.

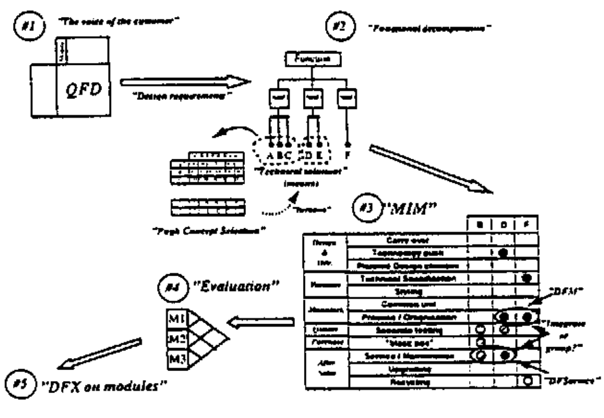

Figure 6: MFD: Modular Function Deployment, (Erixon).

Hansen defines requirements for a computer-based supporting tool for configuration. Configuration in this context means combining sub-solutions to an overall solution. The operations identified by Hansen which the supporting tool must have to enable the designer to handle machine system relations are "define relation", "show relation", "modify relation" and "set flag". The final operation is used to indicate the status of the relation, *i.e.* to enable the designer to identify which relations are solved, and which yet have to be solved.

4 The design data management interpretation

In the data management interpretation, product structure is regarded as the carrier of or the framework for design data. The core of this data is created during the design of the product. Besides the design activity itself, users of the data are all activities in the company: sales, production planning and control, after-sales activities etc. This is also illustrated by a chart from the Ericsson company, see figure 7.[2]

As said before, this interpretation may be distinguished, but not separated from the first interpretation, the design interpretation. One task of design data management is to document the results of design activities, including the product structures. Work done in this interpretation distincts itself from the first interpretation because of a number of reasons:

- the focus is on *data management*, rather than on the creation of data;

[2]The chart was presented during a discussion by Mr. Nyberg, representative from the Ericsson company.

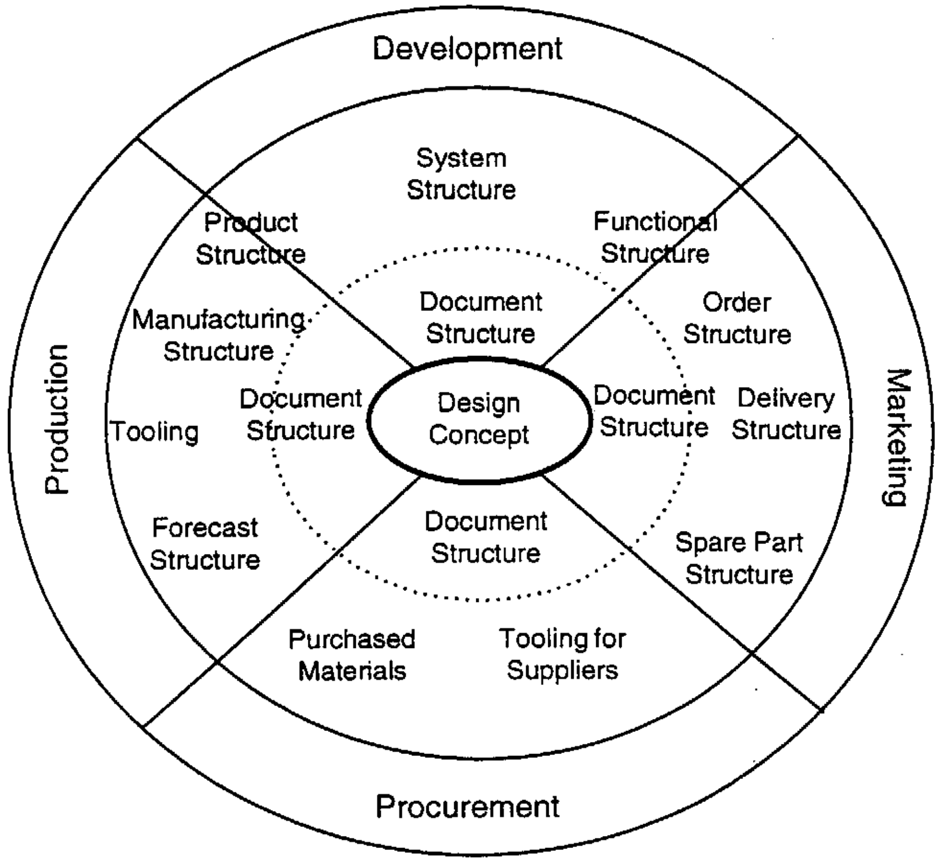

Figure 7: Product views from different perspectives (Ericsson company).

- the focus is in this interpretation on *design data*, which covers product and life-cycle process defining data, and design process data;
- the *users* of the data structures are *not only designers* as is largely speaking the case in the first interpretation. People from sales, design management, production planning and control etc. also use design data.

Work done in this interpretation of product structure shows two main interests, of which in particular the first one received attention during the workshop, see figure 8:

Defining design data frameworks. This issue is concerned with defining the design data that needs to be captured, *i.e.* the data that is generated during design. Related to this is the definition of the use of design data.

Creating support for design data processing. Support is necessary for capturing, retrieving, representing, transforming etc. of the design data.

4.1 Defining frameworks for design data

The essential question here is which data must be captured. Design data covers both product related data and design process related data. Product data covers descriptions of product structures, descriptions of elements

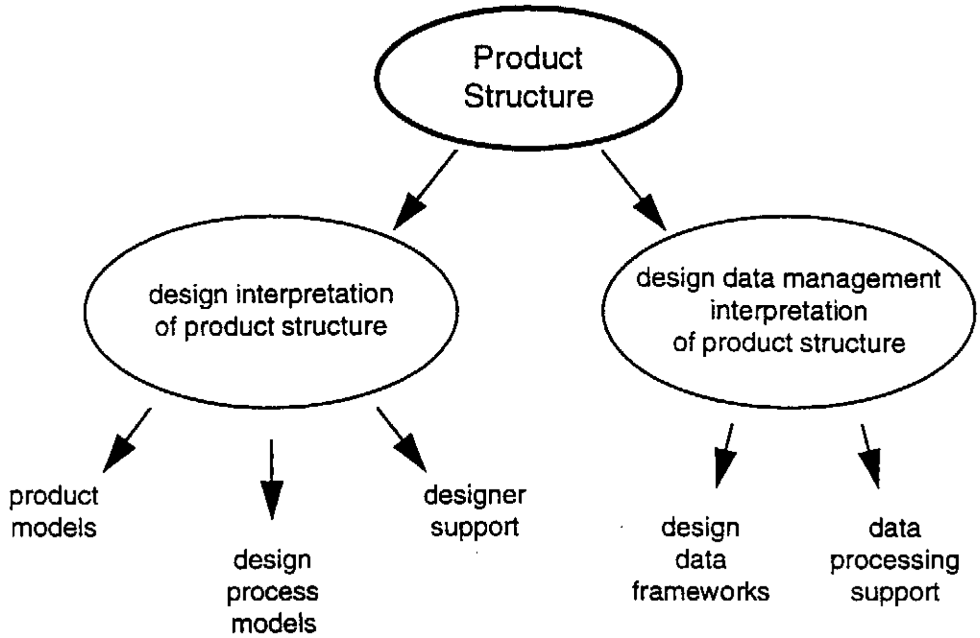

Figure 8: Issues in the second interpretation of product structure.

and relations (geometric data), specifications, production processes, suppliers, etc. Design process related data is information on design rationale (arguments and decisions), the status of the design, the design history, versions of the design, process administration data (*e.g.* planning), etc. Defining design data frameworks requires understanding of the data need of the consuming processes.

The framework described by Van den Hamer is based on the identification of five dimensions. The dimensions are at the same time relevant for data management and for design process management. The five dimensions are (figure 9):

- the *versions* dimension. Versions of a design are the result of modifications. Therefore, related to design versions are iterations in the design process;
- the *views* dimension. A view corresponds to a different aspect of the same design. Identifying and utilizing views allows for reduction of the complexity of the design task. Each view is created in different steps;
- the *hierarchy* dimension. The hierarchy explains consists-of relationships. It is another important mechanism to reduce complexity in design. It also simplifies reuse of subdesigns. On the design process side, hierarchy corresponds to subprojects;
- the *status* dimension. The status expresses the maturity of a design. It corresponds to verification, validation and approval in the design process;
- the *variants* dimension. Variants are relevant for developing families of related products.

Blessing indicates that traditionally only data on the final design is saved; hardly any data on design rationale is stored, the history of development is lost. Capturing more data would increase reuse of former designs and design processes. In developing computer-based designer support, capturing more data is relevant for a computer-system being able to provide "intelligent" advice. The limitations of capturing data are not technical: there are sufficient means to capture any type of data one wants to capture. The limitation is a practical one: "all" data is an enormous amount of data, practically impossible to be described by the designer.

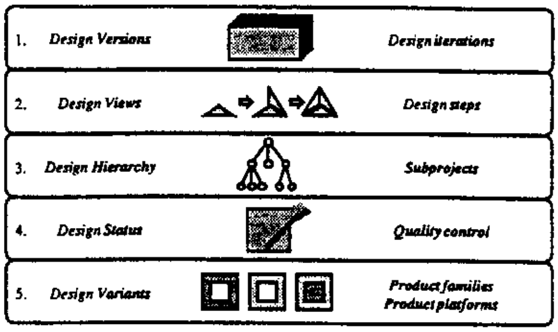

Figure 9: The 5 dimensions of design (van den Hamer).

The core of the PROSUS system (PROcess-based SUpport System) Blessing is working on, is the so-called design matrix, see figure 10. The design matrix provides a framework for supporting designers in their activities. Blessing states that the matrix is not intended to define how designers do design, nor how they should design, but rather how they could design. In an evaluation of the PROSUS system, using a manual implementation of the design matrix, the designers who used the design matrix created more concepts, documented more of their design process, assessed their solutions more often during design and were able to apply various approaches. However, the design process did not become more efficient, which is expected to be due to the far-from-ideal implementation, and the quality of the design did not improve.

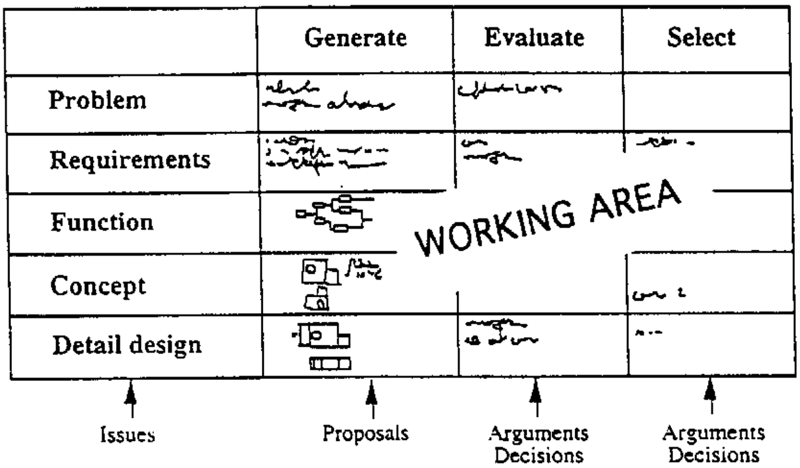

Figure 10: A simplified design matrix (Blessing).

Vroom reports on studies of the information flow in product and process development within three companies (automotive suppliers). The goal of studying the information flow is to improve the development process by optimizing information supply. An optimized organization of the information flow is also a prerequisite for implementing design data management tools like PDM (Product Data Management) systems. Vroom has created a metamodel for storing and representing the studied information flows, see figure 11. The metamodel distinguishes three objects: the subject, *i.e.* the acting system (*e.g.* man or department), the activity carried out by the subject and the information created or used by the subject in carrying out an activity. In this metamodel, eight representations of the information flow can be derived. In the information object, four types of information are distinguished: design rationale, product and process design descriptions, technical documentation and available knowledge, and project management information. Each of these types of knowledge must be managed differently, because they differ with respect to being project specific or not and with respect to being accessible to the entire department or not.

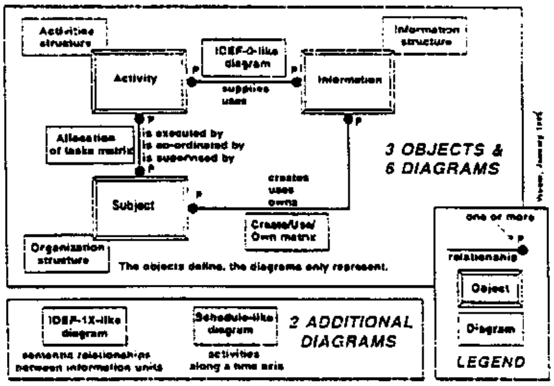

Figure 11: A metamodel for describing information flows (Vroom).

Modelling and managing product knowledge is essential to carrying out design. In her work, Zhang develops a scheme for product knowledge modelling, and proposes a computer-based solution to product knowledge management. Product knowledge consists of specific product knowledge and product domain knowledge, see figure 12. Specific product knowledge is knowledge on a specific design at hand, *e.g.* specifications, or functional or physical product descriptions. Product domain knowledge is generalized knowledge on products. In addition, Zhang defines design process/activity knowledge, which is knowledge on the design process (not shown in the figure).

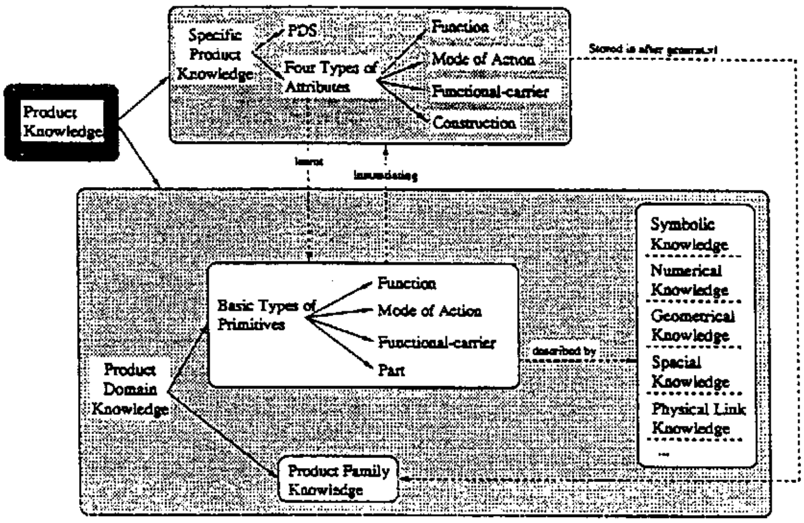

Figure 12: Classification of product knowledge (Zhang).

During the previous workshop, Yu presented work on configuration design [4]. Underlying to the chosen approach is a concept of two interrelated product structures: a Product Breakdown Structure (PBS) and a Product Family Classification Tree (PFCT), see figure 13. A PBS represents a solution. It shows the main system and its subsystems. In configuration design, the solutions for design problems on any level are searched for in the PFCTs. A PFCT contains a group of elements with high commonality, for instance regarding function or manufacturing process. In her contribution to this workshop, Yu reports on an industrial evaluation of the existence of these structures. Both the PBS and PFCT can be constructed, however with some difficulties because of lack of information, poor documentation, high product complexity and limited product knowledge at conceptual level.

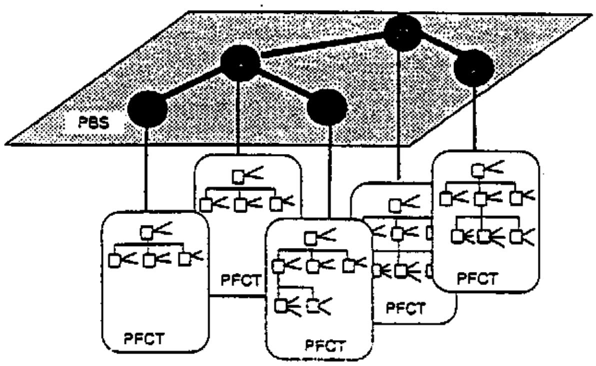

Figure 13: Product Breakdown Structure (PBS) and Product Family Classification Tree (PFCT) (Yu).

Van Holland argues that the traditional product models are not sufficient or inconvenient to support tasks like process planning. Additional information is required, which can be stored in a feature model. Van Holland describes an object oriented data structure for describing products using both form features, *i.e.* features which are used to model single parts, and assembly features, *i.e.* features which are used to model assemblies. Figure 14 shows the class hierarchies for form features and assembly features. In addition, a class feature is defined for constraints, which express relationships between features.

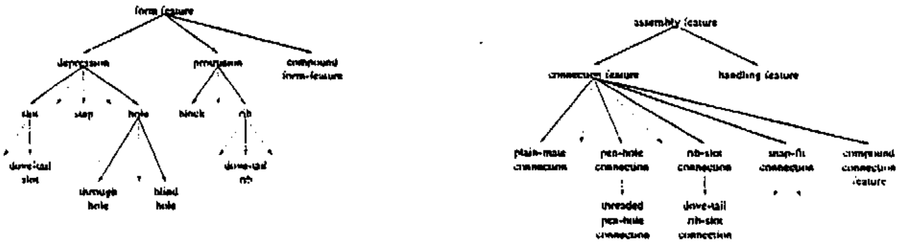

Figure 14: Class hierarchies for form features and assembly features (van Holland).

4.2 Support in data processing

The term data processing covers both the capture of the design data (interface to the data sources) and the presentation of the data to the consumers. This topic was hardly addressed at the workshop. Only Blessing described the development of a system, called PROSUS, which aims at data processing during design. At the first workshop, the topic was only addressed in [5]. Among the required functionality of data processing support are:

- easy (on-line) data capture, indexing and storage;
- fast data browsing and retrieval;
- representation of data to its users;

- data transformation between models/views;
- consistency management.

5 Concluding remarks

This paper has tried to provide an overview of the results of the second WDK Workshop on Product Structuring.

Compared to the first workshop, a major step forward was made in the understanding of the field. The two interpretations of product structure described in this paper seem to be an elegant classification, providing an entry to understand the work done in the field.

Depending on the viewpoint of a user, many structures can be defined for one product. The importance of structuring in particular for managing diversity was shown by several contributions.

Design data frameworks received a lot of attention in this workshop. The various contributions on this topic showed four types of data:

- product design descriptions, *e.g.* functional descriptions, physical descriptions, specifications, ...;
- generalized product information, *e.g.* know-how/expertise on product technologies, classes of standardized solutions, ...;
- design process status descriptions: *e.g.* design rationale, planning data, ...;
- generalized design process knowledge, *i.e.* knowledge on design procedures to be used.

An important question raised during discussions was the practical (industrial) relevance of the work presented. Both interpretations of product structure, as introduced in this paper, have their own industrial questions, see the following listing:

- on the design interpretation of product structure:
  - which product structure views are used by designers?
  - can tools be developed which represent the various structures to designers?
  - can tools be developed which allow designers to analyse the structural aspects of their designs?
  - can the benefits of properly structuring products be quantified?
- on the design data management interpretation of product structure:
  - which data should be stored in industrial practice?
  - can tools be developed which support in this extensive data capturing need?
  - what is the effort involved in data capturing (what does it cost) and what is being gained?
  - which data should be captured for other processes than the design process? How to link these data frameworks to the ones created during design?

Acknowledgment

The authors want to thank Michiel Willemse for discussions on this paper and for proofreading earlier versions of the paper.

References

[1] M Tichem, T Storm, M M Andreasen, and K J MacCallum, editors. *Proceedings of the WDK Workshop on Product Structuring*, 1995. June 22-23. Delft University of Technology, Delft, the Netherlands. ISBN 90-370-0137-8.

[2] M Tichem, T Storm, M M Andreasen, and K J MacCallum, editors. *Proceedings of the 2nd WDK Workshop on Product Structuring*, 1996. June 3-4. Delft University of Technology, Delft, the Netherlands. ISBN 90-370-0148-3.

[3] C T Hansen. An approach to configuring mechanical systems. In M Tichem, T Storm, M M Andreasen, and K J MacCallum, editors, *Proceedings of the WDK Workshop on Product Structuring*, pages 51–69, 1995. June 22-23, 1995, Delft University of Technology, The Netherlands, ISBN 90-370-0137-8.

[4] B Yu and K J MacCallum. A product structure methodology to support configuration design. In M Tichem, T Storm, M M Andreasen, and K J MacCallum, editors, *Proceedings of the WDK Workshop on Product Structuring*, pages 157–175, 1995. June 22-23, 1995, Delft University of Technology, The Netherlands, ISBN 90-370-0137-8.

[5] T Murdoch. A layered framework for structuring product data. In M Tichem, T Storm, M M Andreasen, and K J MacCallum, editors, *Proceedings of the WDK Workshop on Product Structuring*, pages 139–156, 1995. June 22-23, 1995, Delft University of Technology, The Netherlands, ISBN 90-370-0137-8.