Product architecture development enabling integrated re-design of mechanical products

R.E. Begelinger, E. Post, F.J.A.M. van Houten, H.J.J. Kals

Abstract

Global competition forces companies to increase their competitive advantage. The design process represents an interesting area to improve the overall business performance. There are two topics involved in improving the design process. The first one is the integration of constraints from the product life cycle in the design process. The second one is the improvement of the ability to judge the influence of a certain design decision quickly. Both topics are dealt with in this paper by describing the new concept of the Product Configuration Structure (PCS). The PCS is used to represent different product configurations with respect to functions, means, components and production methods in a structured way. The product configuration structure serves as a product information backbone in an integrated re-design tool called PROFIDT. To create a better integration of the flow-based function structure and the hierarchical component structure a new concept of connection functions is introduced.

1 Introduction

As a result of increasing competition, companies are forced to deliver their products with shorter delivery times and better quality at a lower price. Optimisation of one of these performance criteria would not be very difficult but simultaneous optimisation is a difficult task.

According to [Ehrlenspiel 85] 70-80 % of the cost of a product are committed during the design stage. Also the quality and the delivery time are largely influenced by decisions taken in the design stage. These figures indicate that the design stage is the most suitable stage of the product life cycle to optimise these performance criteria.

If the emphasis is put on the improvement of the design process two criteria have to be satisfied:

1. During design, constraints from all phases of the product life cycle should be taken into account. This is called integrated design.
2. It must be possible to quickly judge the quality of a design. The quality of the design depends on how much the product's quality, delivery time and costs are conforming to the customer's expectations.

These two criteria are hard to satisfy when new products have to be designed. However for existing products that have to be modified or re-designed, a lot of a knowledge about it is already available. This offers possibilities to improve the output of the design process.

2 Design of the product architecture

A special case of re-design is the design of customer specific product families. These can be defined as a range of products based on the same *product architecture,* where every instance is directly related to a customer order. A product architecture can be defined as the scheme by which the functional elements of the product are arranged into physical chunks and by which the chunks interact [Ulrich 95].

The design and representation of a generic product architecture, which is capable of representing all different instances of a product family, is one of the key elements in creating a specific re-design support system. The product architecture provides a framework in which information, necessary in the product life cycle, can be contained. This information will consist of price information, manufacturing process data, recycling information etc. A change in the information concerning any phase of the life cycle, or a changing customer specification, will lead to a modification of a product design. The modifications consist of dimensional changes, selection of different components or topological changes caused by selection of a different working principle.

To create a suitable product architecture, which supports all the stages of the product life cycle, it is important to understand the design process of a product and it's specific problems.

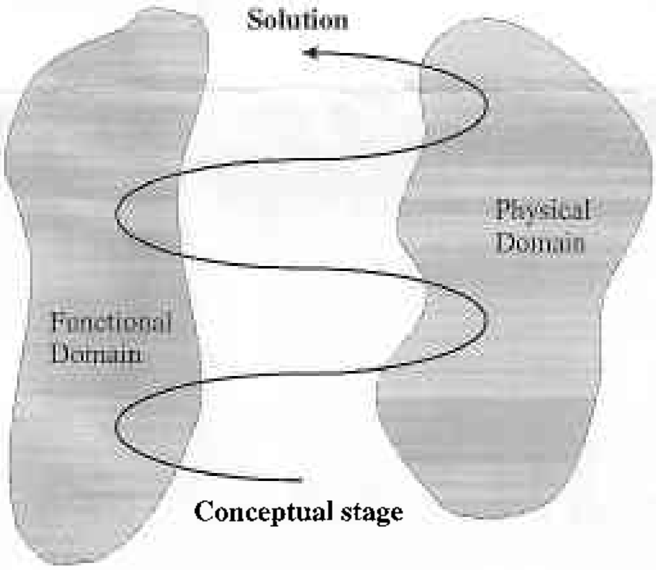


Figure 1: Design as a iterative process.

Design can be seen as an iterative process [VDI 2221] in which some steps are repeated until a satisfying result is reached.

In figure 1 this process is shown. In this process repeatedly a desired function is defined which is mapped on the physical domain by selecting an appropriate working principle or physical element. On a lower abstraction level the chosen working principles can lead to the definition of a sub-function etc. This process continues until all functions have been embodied in physical components. The design decisions at conceptual level can be seen as starting points for design branches. The correctness of a chosen design branch can only be judged when the branch is detailed down-to the component level. As stated in the introduction, it is essential to be able to perform this process quickly. Most companies will stop the design process after the evaluation of the first design branch that leads to an acceptable result. This is not necessarily the best one. We will show that the product architecture described in this paper provides designers with a tool to evaluate a large number of design branches, ensuring a better solution.

To accommodate future design changes, it is necessary to represent the above described design process in the product architecture. This means that both the functional structure and the physical structure need to be integrated into the product architecture. The differences between the functional and the physical structure lead to problems in creating the product architecture. In the functional domain functions can be found by analysing the flow of energy (E), information (I) and material (M) [Kroonenberg 83][Ullman 92]. This leads to a functional structure like a flow chart, connecting output from one function to the input of another, while on the other hand the physical structure is a hierarchical structure representing the decomposition of the product into sub-assemblies and components.

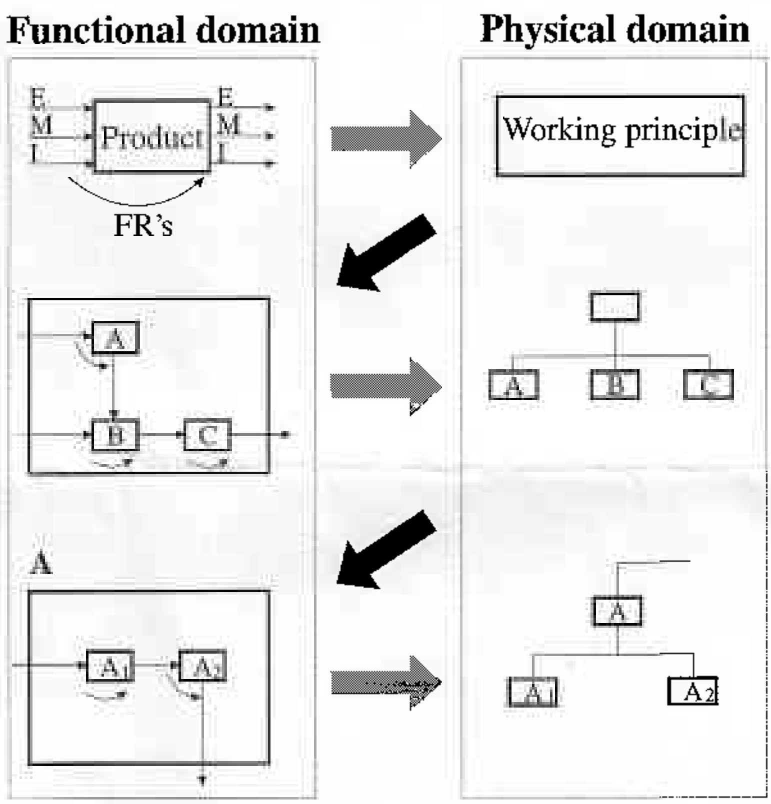


Figure 2: Mapping between functional and physical domain.

Figure 2 shows the functional and physical domain and the mapping between them. The functional requirements (FR) define the overall function which has to be performed.
In the physical domain this required function is then transformed into a working principle. On a lower abstraction level the working principle can be divided into a number of sub-functions in the functional domain (stepwise refinement). The last step will be the selection of components to fulfil the working principles, which leads to a hierarchically structured technical system consisting of assemblies, sub-assemblies and components.

Another complicating factor in creating a product architecture is that the mapping between the domains is not a 1:1 operation. For instance, several components can fulfil one function or one component can support a number of functions. An example is a nail clipper assembly by Ulrich shown in figure 3. In this example a nail clipper is designed in two different ways. In the first design, a 1:1 mapping of function and means is accomplished. This leads to a large number of components and therefor to a difficult assembly sequence. In the second design, certain functions are integrated into one component, resulting in far less components.

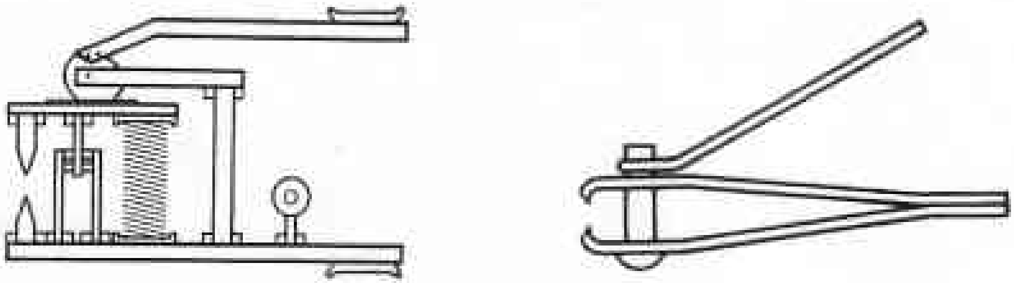


Figure 3: nail clipper assemblies (redrawn after Ulrich)

This example shows that integration of multiple functions into one component creates a better design as long as the integration of components is performed in accordance with the first design axiom of Suh [Suh 90]. This axiom states that the independence of functional requirements should be maintained. Often the mapping from number of functions to number of components can vary, based on the lot size.
For large lot sizes it is possible to invest in product tooling (die casting, forging, injection moulding). This enables the economic creation of complex geometry, as a result of the integration of several functions into one component. For small lot sizes it is not possible to make large investments in specific tooling. This leads to the use of components of which the geometry can be created with standard production tools. So the lot size influences the mapping between functions and components.

The next section describes the way to incorporate the three properties essential for an appropriate product architecture as elucidated above:

- stepwise refinement (functions-means) to capture design intent/history
- integration of function structure (flow) and means structure (hierarchical)
- representation of different mapping possibilities (number of function to number of components) based on production method

3 Product Configuration Structure

One of the main objectives of a product architecture is to provide a framework to store and retrieve information about the life cycle of a product. Models of the design synthesis process use the basic elements functions and working principles (means) to describe the iterative nature of the design process [Hubka 84] [Andreasen 87]. Tjalve [Tjalve 78] introduced the "Funktion-Mittel-Baum" (Function-Means-Tree) which we use as a basis to describe the product architecture instead of the design process itself. We extended this structure by adding models of components to the functions and means. In this way, a difference between the abstract means and the physical embodiment is created. As a result of these extensions it is possible to fully implement the theory mentioned before in a software tool. In our opinion this is the only way to use this kind of design theory for practical purposes

In the "function-means tree" theory no difference is made between functions which are essential for the performance of a device (principal functions), and functions which connect the principal functions. In our Product Configuration Structure (PCS) this difference is made. The connection of principal functions is done by so-called connection functions. These connection functions represent the *flow* of energy, material and information from one principal function to another, whereas the principal functions represent the *transformation* of energy, material and information. This way a better connection is made between the functional flow structure in the functional domain, and the hierarchical structure in the physical domain. The embodiment of both principal functions and the connection functions is assumed to be ideal in this stage. In practice, components performing a connection function often also transform energy as a result of unintentional secondary function behaviour, e.g. energy loss due to friction. In this paper the design of a product architecture using ideal components is described. Future work will incorporate the recognition and modelling of non ideal behaviour of components. These components can extend the information in the product architecture by describing the secondary function behaviour like, for example, energy losses. This information can be used for tolerance analyses and simulation of product behaviour.

In the previous section three properties of a product architecture have been described. The first two have been discussed above. The last one is the need to incorporate production method information in the product architecture, and to show it's influence on the mapping between the number of functions and the number of components. This problem has been dealt with by the introduction of a "production method level" in the function-means tree. This level, situated between means and component level, creates a design branch based on the production method. It enables the integration of process planning in the product architecture. As a result, it is possible to evaluate different production methods for a component based on the production capacity available.

Figure 4 shows the resulting PCS.

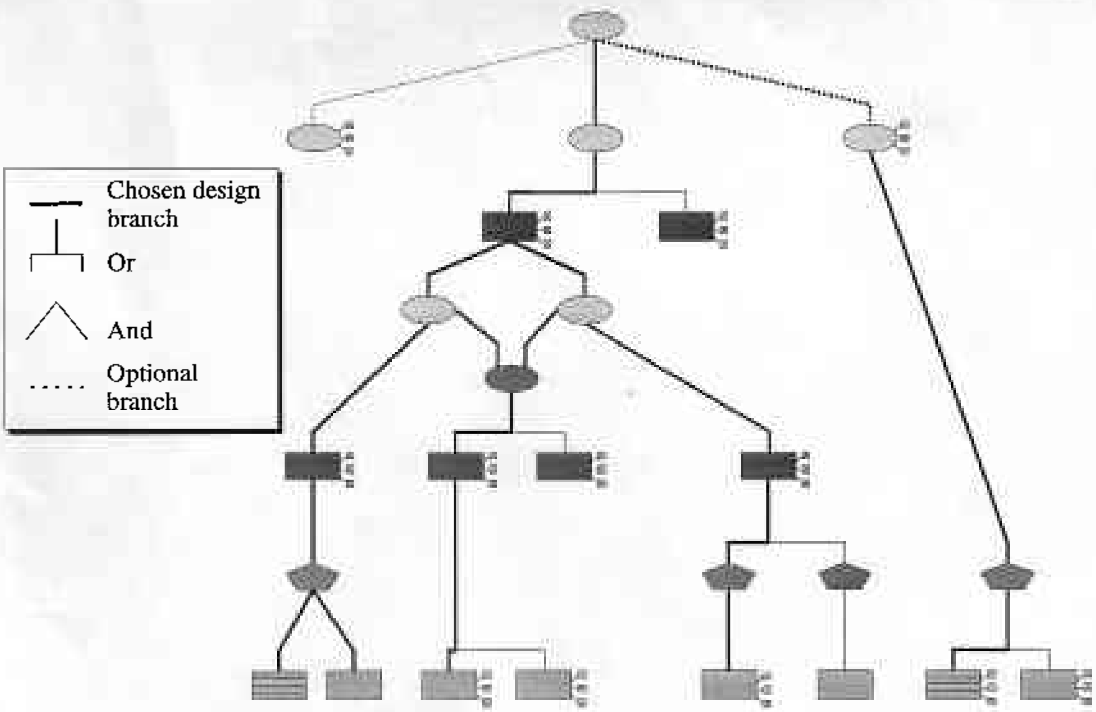
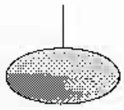


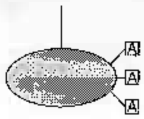
Figure 4 : The Product Configuration Structure.

The PCS shows different design branches which all can lead to a feasible solution to the design problem. By means of a computer support tool, these branches can be evaluated quickly in order to make the right design decisions. By comparing design parameter values with attribute values, associated with alternative options, an appropriate option will be chosen. The parameters will get their value automatically or by user input. If a technological relation constrains the choice between alternative options to just one option it will get it's value automatically. If multiple options are technologically possible, the designer can evaluate the consequences of each possibility and select the best one (See figure 5). The selection of alternative options can also be a result of direct input of customer specifications.

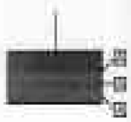
The different symbols used to represent the PCS will be elucidated.



Function
This class represents a principal function. Principal functions transform material, information or energy. The relation between the input situation and the output situation is the definition of the function.

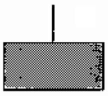


Optional Function
This class represents optional functions. Optional functions are functions that are not necessary for the support of the main product function. Optional functions can be added if the customer is willing to pay for them.
An example of an optional function is a device to monitor bearing conditions. The absence of this device will not affect the principle function of the product. Because of the fact that optional functions are functionally independent they can be offered in separated modules.

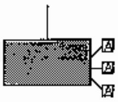


Means
Means are the abstract elements which are chosen during the design process. Often there are different feasible means to fulfil a required function. These means contain attributes, like material, manufacturing etc. which make it possible to select depending of the requirements and constraints. An example is the fulfilment of the function to transform mechanical energy into fluid energy by either a centrifugal pump or displacement pump which have a different working principle.

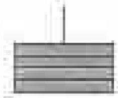
Production method
Often it is possible to choose between different production methods to create a component. By the selection of the production method, the architecture of the product will be influenced. Components can, for example, be milled or injection moulded. The various options are represented in the product architecture by means of the production method class.



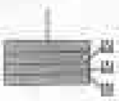
Component
Components represent the lowest level in the hierarchical structure of a product. They are the basic physical elements in a technical system.



Alternative Component
To fulfil a certain function, a selection between alternative components has to be made. The selection between alternative components can be accomplished by comparing their attributes to the desired ones. For product families, alternative components mostly consist of size ranges.



Combined Component
A combined component fulfils a desired means in combination with one ore more other components. An example is the use of a bolt and nut to fulfil a connect means. Combined components can also be the result of a production method. For example, a welded construction can be a combined component while it's cast counterpart consist of just one component



Alternative Combined Component
Alternative selection of combined components based upon different attributes. For example the selection from a size range of welded components.

Connection
This type of elements connects functions, means (abstract connects) and components (physical connects). These connections represent the *flow* of energy, material and information from one element to another. When the technical system is fully designed, including the embodiment, connects will only connect components to components.

The above mentioned elements make it possible to integrate the three quality's discussed in section 2.

4 Implementation and conclusions

The Product Configuration Structure as described in the previous section has been integrated as an information backbone in a prototype integrated re-design support system called PROFIDT. An implementation has been carried out for BW/IP International, a world wide operating company producing pumps, valves and mechanical seals [Begelinger 97]. In this implementation a commercial 3D parametric CAD system has been used for the automatic creation of the geometry, based upon the input generated by PROFIDT [Post 97]. The pump family which has been implemented is a typical customer specific product family. This means that all of its members can be represented with one PCS. Figure 5 shows part of the PCS showing two different design branches for the drive function. The first branch results in a directly driven configuration while the second one results in a configuration driven by an E-motor combined with a gear box. The corresponding CAD assemblies are shown on the right side of their PCS.

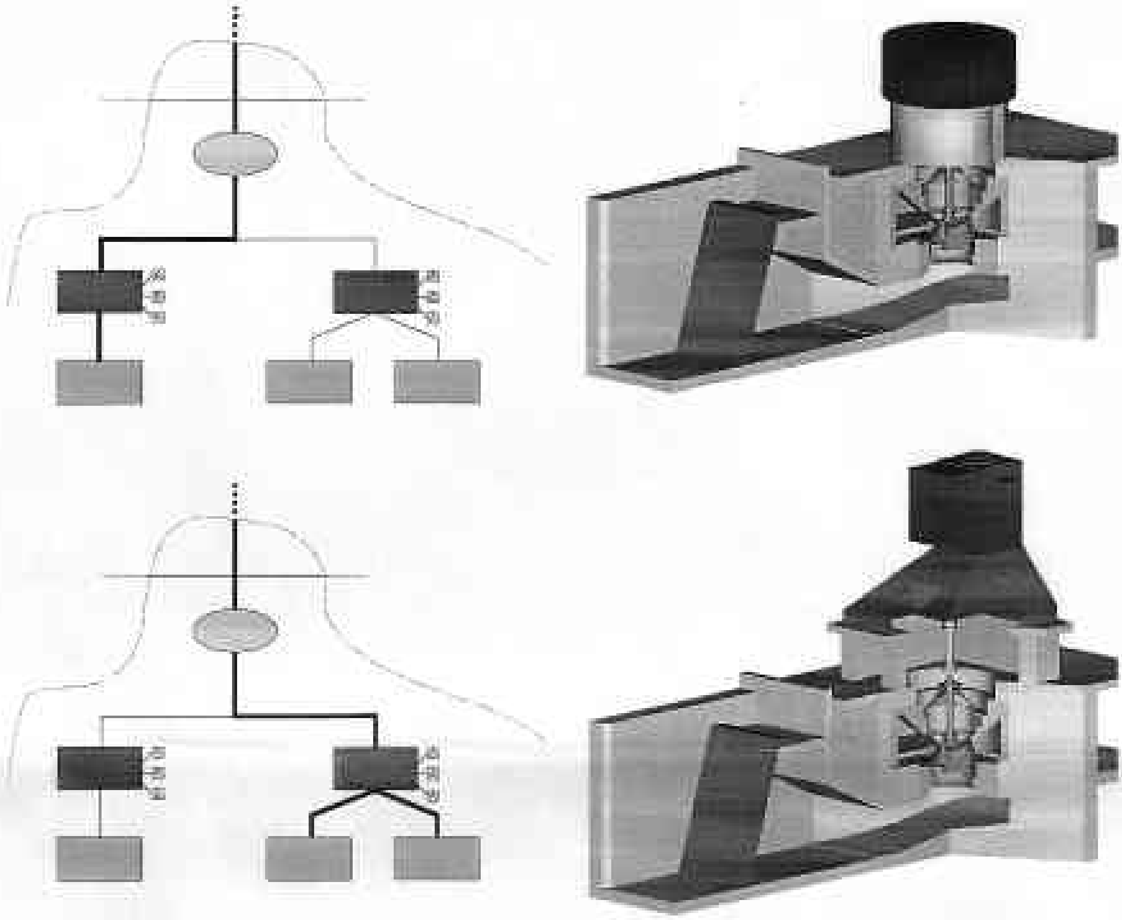


Figure 5: PCS with corresponding CAD assemblies for two different configurations

The system is operational in the company since early 1997. The results look promising. With the aid of the system several feasible design solutions can be evaluated quickly. This way the product design can be optimised, within the limits of the PCS, with respect to delivery time, quality and price.

Literature:

[Andreasen 87] Andreasen, M.M., Hein, L., Integrated product development, Springer-Verlag, Berlin, 1987, ISBN 3-540-16679-3.

[Begelinger 97] Begelinger, R.E., Computer Aided Design of customer specific products based on product families, de constructeur September 1997, Samson bedrijfsinformatie, Alphen a/d Rijn, page 28-31, ISSN 0010-6658.

[Ehrlenspiel 85] Ehrlenspiel, K., Design for cost (in German: Kostengünstig Konstruieren), Springer-Verlag, Berlin, 1985, ISBN 3-540-13998-2.

[Hubka 84] Hubka, V., Theory of technical systems, Springer-Verlag, Berlin, 1984, ISBN 3-540-17451-6.

[Kroonenberg 83] Kroonenberg, H.H. van den, Methodisch Ontwerpen, course notes, in Dutch, University of Twente, Enschede, 1983.

[Post 97] Post, E., Development of a Parametric Design system for BSV/BCV pumps, Internal report, University of Twente, Enschede, 1997.

[Suh 90] Suh N.P., The principles of design, Oxford University Press, New York, 1990, ISBN 0-19-504345-6.

[Tjalve 78] Tjalve, E., Systematische Formgebung für Industrieprodukte, VDI-Verlag, Düsseldorf, 1978, ISBN 3-18-419053-6.

[Ullman 92] Ullman, D.G., The mechanical design process, McGraw-Hill, New York, 1992, ISBN 0-07-065756-4.

[Ulrich 90] Ulrich, K.T., Seering, P.S., Function sharing in mechanical design, Design Studies, Vol. 11, No 4, October 1990, ISSN 0142-694x.

[Ulrich 95] Ulrich, K. T. and Eppinger, S.D. Product design and development. McGraw-Hill, New York, 1995, ISBN 0-07-113742-4.

[VDI 2221] Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. Düsseldorf: VDI-Verlag 1993.