INTERNATIONAL DESIGN CONFERENCE - DESIGN 2002
Dubrovnik, May 14 - 17, 2002.

A CLASSIFICATION OF MATRIX-BASED METHODS FOR PRODUCT MODELING

Johan Malmqvist

Keywords: Matrix methods, Product models, Design research

1. Introduction

Current products are becoming increasingly complex. In the development of complex products considerations of the product structure and relations within the product are crucial. Many aspects need to be considered, including business aspects, such as commonality, as well as technical aspects, such as the geometric interfaces between the components of the product.
Matrix-based product modeling methods are increasingly being used in such work. The most common aim is to facilitate analysis of relations in complex systems, supporting for example

- product modularization [Erixon 1998],
- analysis of technical interactions in products [Pimmler & Eppinger 1994],
- design analysis [Suh 1990], and
- change impact analysis [Clarkson *et al*. 2001].

However, there exists no classification of such methods. Such a classification could be used to

- support the selection of a method for use in a particular problem situation
- direct future developments by showing opportunities for new developments
- position different methods with respect to others, including identification of overlaps

The goal of this work is to develop such a classification. The research approach used combines the results from a survey of existing methods and from a theoretic analysis, based on the technical systems theory [Hubka & Eder 1988]. The resulting classification is itself organized as a matrix, where the rows and columns are derived from the technical systems theory with some extensions, i.e. including properties, functions, organs, components, life-cycle processes and product variants. A large number of matrix methods are then positioned using this classification matrix.
The remainder of the paper is structured as follows. In section 2, we provide definitions for the basic variants of matrix-based modeling methods and introduce a initial, scope-based, classification of matrix methods as being element-level matrix methods, product-level matrix methods and matrix-based methodologies. Sections 3 and 4 provide a more detailed analysis of what product element types and relations that can be included in a matrix-based product model. In section 5, a content-based classification of matrix-based methods is presented, and ideas for future work are suggested. Finally, section 6 lists conclusions.

2. Matrix-based product modeling methods

This section aims to discuss what a matrix-based product modeling method is, and to classify such methods according to their scope.
In our view, a matrix-based product modeling method represents some view of the product structure (product elements and their relationships), shown as a matrix. This view is used to support analysis of the product function, interfaces, assembly etc. Depending on application, various algorithms such as clustering, partitioning, coverage can be used to manipulate the matrix. Many matrix-based product

modeling exist. A representative sample is referenced in the text and is included in the reference list. The spatial constraints of this paper do not permit an extensive review. However, key contributions are included.

We will in the remainder of the paper use the acronym P-DSM (Product modeling Design Structure Matrix) for such methods. This is motivated by that conventional DSMs [Steward 1981] constitute a major part of the methods that we are interested in. However, conventional DSMs are also used to analyze tasks and organizations and are defined as having the same elements in the rows and columns (inter-domain matrices, see below). Here, we focus on product DSMs only and also consider matrices with different types of elements in the rows and columns (intra-domain matrices). Hence, the modified acronym. We begin by discussing element-level P-DSMs and continue with product-level P-DSMs and matrix methodologies. The findings of this section are summarized in figure 1.

2.1 Element-level P-DSMs

Element-level P-DSMs represent the relationships between the elements/parts/components of a single product in a matrix. There are two sub-types of element-level P-DSMs: inter-domain P-DSMs and intra-domain P-DSMs.

Inter-domain P-DSMs represent relations between elements of the same type, for example between two components. The matrix representation can be applied on various levels of abstraction, from properties to components, as shown in figure 2, where the inter-domain P-DSMs are situated along the diagonal of the matrix. The figure further shows that inter-domain P-DSMs can be applied to study interactions between properties/requirements, between functions, between sub-systems/organs/design parameters, between components, and between the life-cycle systems/processes related to the product.

Intra-domain P-DSMs represent relations between elements of different types. As shown in figure 2 (all cells except the diagonal) these are usually mappings from a more abstract element type such as function to a more concrete, such as a sub-systems that realizes the functions. The matrix can then be described as representing a set of design decisions or relations between what and how. One example is the Axiomatic Design Matrix [Suh 1990], which captures the relationship between a property or a function, and the design parameter that realizes the property or function. Another example is the Quality Function Deployment (QFD) [Akao 1990] matrix that shows the relationships between a component and the processes that are used to manufacture it.

2.2 Product-level P-DSMs

Product-level P-DSMs provide a mapping between a set of properties or other elements and a number of “whole” alternatives rather than parts. The motivation for such methods is to support decision-making about entire products or product platforms, of which product-level variants are the “parts”. For instance, developing car platforms requires considerations of common and unique modules within a brand and within a platform but also to see each car variant as a whole [Sudjianto & Otto 2001]. A matrix representation of a platform can then serve as an architecting tool. Examples include the classic methods for concept screening and scoring [Ulrich & Eppinger 2000]. Recent developments have focused analysis of product platforms [Dahmus *et al*. 2000, Sudjianto & Otto 2001]. The product-level P-DSMs that we have found have all been intra-domain. Exploring inter-domain product-level P-DSMs could be an area for future work.

2.3 Matrix-based methodologies

Matrix-based product modeling methodologies utilize some set of P-DSMs in a systematic and coherent way in order to manage a multi-dimensional, complex problem. In addition to the support for solving particular problem, the use of matrices can then be seen as a way of thinking (a mindset) or a standardized mode for communication and documentation.

One category of such methodologies has a strong prescriptive element and aims to support the entire development process from identifying customer needs to establishing manufacturing processes. Examples include QFD [Akao 1990, Clausing 1994] and Axiomatic Design [Suh 1990, Vallhagen 1996]. Various matrices are here used throughout the process, at least to support the major steps. Figure 2 shows how QFD provides intra-domain matrices for most levels of abstraction, but also that the platform and organ levels are not considered by this methodology.

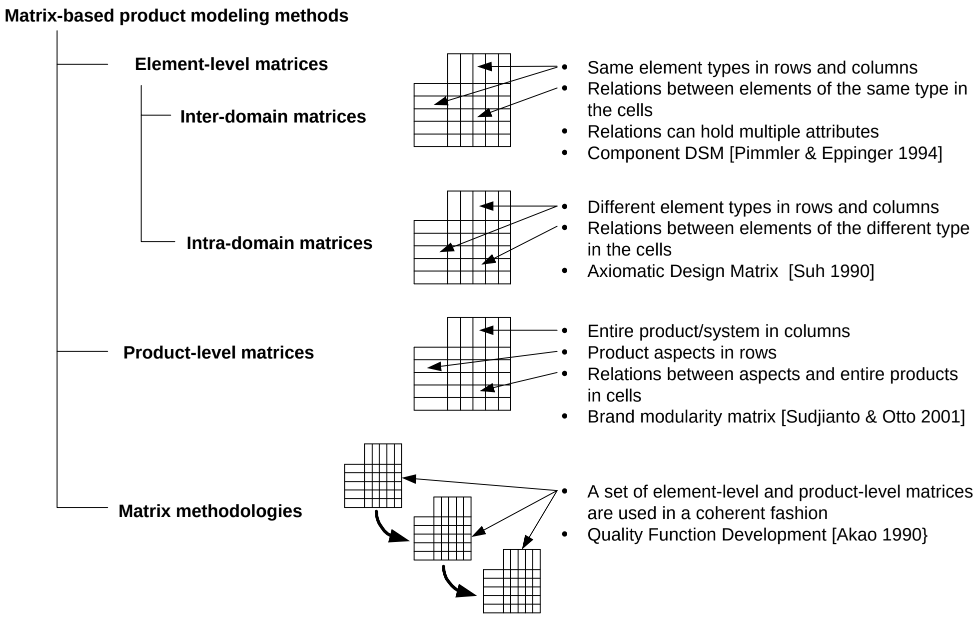

Figure 1. Overview of matrix-based product modeling method types

Another category uses matrices as a framework for modeling complex, multidisciplinary systems. For example, product data management (PDM) system development requires consideration of process, information (product), organization and systems [Malmqvist & Svensson 1999], and inter- as well as intra-domain relations need to be taken into account. Existing methods can effectively model some aspects, but it is difficult to get an overview of the entire PDM system. Using a matrix-based framework then facilitates switching between perspectives and eliminates redundant data collection. Another example in this category is the framework developed by Eppinger & Salminen [2001] to analyze the interactions between the product architecture, development process and organization.

3. Product model elements

The aim of this section is to identify what product model element types that can be the base for a matrix-based product modeling methods. Theses element types will constitute the span for the content-based classification presented in section 5.

Generally speaking, product models need to cover the entire spectrum from representing customer needs to the points in a geometric model. However, we will here make the delimitation to methods that support design decision-making or interface analysis. Consequently, the finest level of granularity considered is the feature/design parameter/functional surface level.

For product models, the technical systems theory [Hubka & Eder 1988] constitutes a key defining framework. At the core of their theory is the definition of function, organ and component (structure) element types, along with the observation that each of these can be described using several levels of abstract and detail. These element types are also used in many matrix-based product modeling methods. However, some other types are also needed: For support throughout the development process, we also need to add a property element type, to represent requirements and behaviour. Further, models of the life-cycle system/processes that the product "meets" during its life-cycle need to be included, as these meetings determine many of the product properties, for example cost [Mortensen & Andreasen 1996]. Finally, we also need element types that represent whole product or product alternatives, in particular since many product-level matrix methods have the aim of support product platform decision-making.

Summarizing this analysis, the following types of element can be included in a matrix-based product modeling method:

- **Properties**: requirements (desired properties) and behaviour (actual properties). Example: the mapping between customer and engineering requirements in the House of Quality [Akao 1990].

- **Functions**: what the system should do in terms of transforming an input to an output or creating an effect.
- **Subsystems/Organs/design parameters/features**: the entities of the product that realize the properties and functions. These can be subsystems, parameters/dimensions, functional surfaces and so on. A structure of such elements constitutes a function-oriented decomposition of the product. Example: for modeling technical interactions between sub-systems based on a function-oriented product decomposition [Pimmler & Eppinger 1994].
- **Components**: the physical parts of the product, as an assembly-oriented decomposition. Examples: for modeling assembly relationships, for analyzing change propagation [Clarkson *et al*. 2001], and for analyzing strategic relationships such as similar module drivers [Blackenfelt 2001].
- **Life-cycle systems/processes**: the systems that the product interacts with during its life-cycle, including parts manufacturing, assembly, distribution and so on. Example: Schlüter [2001].
- **Product-level alternatives or variants**

Existing methods will be classified according to these dimensions in section 5.

4. Product model relationships

The aim of this section is to identify what product model relationship types that can be represented in the cells of a P-DSM, show how they can be modeled, and to identify analysis methods that operate on the relationships. Please note that these relationship types and modeled attributes are those that were found to have been included in P-DSMs. Other relationships types and models certainly exist, for example shape and material definitions of interfaces, but no examples of their use in P-DSMs were found.

4.1 Relationship types

Concerning relationship types, three main types can be identified in the P-DSM literature: functional, design intentional and strategic. These main types all have sub-types.

4.1.1 Functional relationships

This type relates to an interaction in the system during its life-cycle. These relations are usually purposeful and intentionally designed into the system but we also need to consider incidental relations, for example harmful effects that we have not been able to avoid in the system [Liedholm 1999]. Sub-types of this type include:

- **Transfer relationships**. These relationships show a transfer of material, energy or information in the system that actively and purposefully contributes to the function of the system or is incidental.
- **Structural relationships**. Relationships that exist to maintain the structural integrity of the system.
- **Passive relationships**. Relationships that do not contribute to the active function of the system but are still purposeful, for example protective measures.
- **Mechanical degrees of freedom relationships**.
- **Spatial and positional relationships**.

4.1.2 Design intentional relationships

This relationship type captures different design decisions made in the development process and whose implications one might want to investigate or communicate. Sub-types include:

- **Function-solution relationships**. Relationships that show what sub-solutions that contribute to a particular sub-function. See for example Suh [1990].
- **Selectional relationships**. These show what alternative that has been chosen from a set of alternatives, for example what solution that has been chosen for a particular function in a product-level variant.

- **Decompositional relationships**. The relationship shows the distribution of something on different elements, for example a weight budget or the allocation of functions to sub-systems.
- **Behavioural relationships**. The (actual) relationship between a product component and a product property, for example weight.

4.1.3 Strategic relationships

This type describes the relationship between a particular part and some business-strategy objective, for example standardization or carry-over [Erixon 1998].

4.2 Relationship modeling

The relationship types identified above can further be modeled in many ways, summarized below:

- **As existing/non-existing**. The existing relationships are marked in the matrix, typically with an "X" or a "1". The other cells are left blank, or marked by zeros ("0").
- **Descriptive text**. Variants include functional descriptions ("This interface exists in order to transfer electrical energy from component X to Y") and classifications ("purposeful - incidental").
- **Selections**. A finite set of alternatives can be chosen. For example, alternative solutions for a particular sub-function in a product-level alternative.
- **Qualitative scales**. The strength of the interaction is assessed using a qualitative scale. Examples: "Weak – Strong – Very Strong", "Harmful – Beneficial".
- **Quantitative scales**. The strength of the interaction is assessed using a quantitative scale, possibly measuring a physical property. Examples: "-9 -- +9", "- -- 0 -- +", "F = 10 N". The value for the relationship can be subjectively assessed or computed.

4.3 Analysis methods

Based on the contents of the relationships, a number of methods can be applied to analyze and manipulate P-DSMs. We have identified the following main types:

- **Clustering** methods organize the matrix in such a way that elements with strong relationships are gathered in blocks or "chunks" and the matrix will have a block-diagonal form [Pimmler & Eppinger, 1994, Sosa *et al*., 2000]. The analysis can support tasks like identifying beneficial and detrimental relations. Various indices support the analysis [Blackenfelt 2000]
- **Partitioning** methods sort the matrix in such a way that feedback/iterations in a process are minimized. This can be applied to functions or to design parameters [Suh 1990, Smith & Eppinger 1997].
- **Coverage** methods analyze the matrix in order to detect requirements that are not allocated, functions that are not realized, components that lack an assigned function and so on. Furthermore, potential conflict areas can be identified, for example when a particular function is realized by multiple components [Akao 1990, Clausing 1994, Suh 1990, Malmqvist & Svensson 1999].
- **Index computation** methods compute some aggregate value based on the contents of the cells. Examples include the computation of total value in the concept scoring method [Ulrich & Eppinger 2000] or the strength of a particular module driver in Modular Function Deployment [Erixon 1998].
- **Interaction** analysis methods consider the contents of individual relations and advise strategies for re-design in order to eliminate or at least manage harmful effects [Liedholm 1999].
- **Change propagation** analysis methods follow the relations from a particular element to its closest related elements, and then to other related elements. In this way, the impact of a change proposal can be identified. Aspect such as probability and amount of re-work can be used as factors in the analysis [Clarkson *et al*. 2001]
- **Alignment** methods compare the contents of two related matrices, such as a product and the organization structure. Differences are highlighted in order to identify areas where the

organization may find difficulties in managing interfaces existing in the product [Eppinger & Salminen 2001].

In addition, many matrix methods rely on less formalized analysis methods, such as colour coding.

5. Classification of matrix-based product models

This section will summarize and discuss the results from sections 2-4. The summary is made in form of a matrix, based on the elements of the technical systems theory. See figure 2.

As evident from the figure, the matrix is well populated above and on the diagonal. This indicates that most P-DSMs have been developed to support reasoning either going from a more abstract representation to a more concrete, or within a domain. Only one method goes in the opposite direction, Olsson [1976]. The figure further shows that current P-DSMs cover a major part of what is theoretically needed according to the technical systems theory, with a single but major exception of product-level variants.

The classification matrix can provide a means for selection of methods in a particular situation. In each cell, there are typically several methods. In some cases, these differ with respect to aim and contents. For example, in the Component-Component cell, the strategic DSM includes business issues while the component DSM does not. However, in the Function-sub-system cell, the axiomatic design matrix and the QFD matrix 2 are essentially the same. The matrix presented here allows the detection of such overlaps.

As can be expected, the matrix methodologies provide methods for most intra-domain steps. For example, QFD supports the analysis within the property domain, when going from properties to functions, from functions to components, from components to life-cycle processes and within the life-cycle domain. However, the organ and product platform level is not addressed. Similarly, axiomatic design does not support platform design.

One method is placed in two cells: the axiomatic design matrix. It can be used for mapping from either properties or functions to design parameters. This reflects the flexibility/ambiguity of this method.

A non-named method has been classified as “used in some companies”. This refers to a requirement-part mapping used in at least one Swedish firm. However, we have not found any scientific paper on the pro’s and con’s of this matrix, which shortcuts the contents of several matrices used in the literature.

The content classification matrix has some empty cells also above the diagonal:

- There is a lack of mapping from requirements to life-cycle systems/processes. However, the Olsson matrix [1976] could be used for collecting the same information.
- There is a lack of mapping from function to life-cycle systems/processes. Such a method might be developed, although its application is unclear.
- There are no matrix methods for mapping from sub-systems/organs/design parameters to product-level alternatives. Such a matrix method could be developed to support product architecture design. Sudjianto & Otto [2001] have developed a similar method with function as a basis. A sub-system-based method might have the advantage of easier utilization of product data that exists in a company. Closely related to this could be a matrix that maps from components to product-level alternatives.
- Matrix methods have not yet been applied for mapping between product-level alternatives. If such a method were developed, a major challenge would be the selection of data in the cells. A measure of similarity, perhaps multi-dimensional, could be one alternative.

One issue with P-DSMs is that while they are an excellent tool for visualizing complex relationships and for identifying problem areas, they provide decision support rather than decision automation. The only P-DSM methods that provide a direct recommendation for a decision are the concept screening and concept scoring methods [Ulrich & Eppinger 2000] and the Axiomatic Design Matrix [Suh 1990]. However, major issues that are dealt with by using P-DSMs are less well supported. Examples include platform decisions and function allocation. Here, the P-DSMs show the structure and relations but interpretation is difficult. The problem is emphasized when multiple attributes of the relations need to be considered, for example geometry, function, commonality and so on. There is a lack of measures that combine various attributes into aggregate measures to which clustering algorithms can be applied, or algorithms that can analyze multiple attributes. Some work is on-going in this direction, e.g., Blackenfelt [2001], but more needs to be done.

One major P-DSM application could be change impact analysis. Methods such as that developed by Clarkson *et al*. [2001] are available. However, it may be very difficult to find the input data for the change impact analysis, especially if this type of analysis is going to be performed on a day-to-day basis. Many companies today do not store interface information in a systematic way. Moreover, if the data exists, it is likely to be spread in various source systems such as M-CAD, E-CAD, software code files and so on. There is a need for methods for gathering the data and putting it together. The PDM system may be a suitable place for long-term storage.

	Properties/ Requirements	**Functions**	**Sub-systems/ Organs/ Design Param's**	**Components/ Modules**	**Life-cycle Processes**	**Product-level alternatives/ variants**
Properties/ requirements	QFD matrix 1: roof [1, 5] Trade-off matrix	Axiomatic Design CA-FR matrix [23]	Axiomatic Design FR-DP matrix [23] Optimization problem linearization [2]	Used in some companies Module Indication Matrix [3, 7, 12, 21]		Concept screening [24] Concept scoring [24]
Functions		Task DSM [22]	Axiomatic Design FR-DP matrix [23] QFD matrix 2 [1, 5]	Function allocation (sharing) matrix [11] Tol chain matrix [9]		Modularity matrix [6] Brand modularity matrix [19]
Sub-systems/ Organs/ Design Param's			Parameter DSM [17] Component DSM (func view) [10, 11, 15, 18]	Organ-component mapping [11]	Axiomatic Design DP-PV matrix + extensions [23, 25]	
Components/ Modules/				Component DSM (assy view) [10, 11, 15, 18] Strategic DSM [3] Product risk matrix [4]	QFD matrix 3 [1, 5]	
Life-cycle Processes	Criteria matrix [14]				QFD matrix 4 [1, 5] Life Cycle Design Matrix [16]	
Product-level alternatives/ variants						

Figure 2. Content-based classification of matrix-based product modeling methods

6. Conclusions

In the development of today's complex products, issues concerning design of the product structures are crucial. Matrix-based product modeling methods can support this activity by, for example, visualizing, structuring and analyzing complex products. Moreover, matrix-based methodologies bring together several matrix-based methods to address complex problems. However, most matrix-based modeling methods offer decision support, but not decision automation, and interpretation can be difficult.

Matrix-based methods can be classified according to scope and content.

With respect to scope, element-level matrices, product-level matrices and matrix methodologies can be distinguished. The matrices can be further categorised as inter-domain or intra-domain matrices.

With respect to content, a classification can be developed on the basis of what element types and relations are captured in the matrix. Concerning element types, it is shown that current matrix-based product modeling methods include properties, functions, organs, components, life-cycle processes/systems and product-level alternatives. The relations modeled in the matrices include functional, design intentional and strategic relations. A matrix containing these elements and relations can be constructed in which a large number of matrix methods can be positioned.

Future work in the area should lead to the development of additional methods that support platform development, algorithms that consider multiple attributes per relation and links to external software, for example CAD and PDM.

References

Akao, Y. "Quality Function Deployment, QFD - Integrating Customer Requirements into Product Design". Productivity Press, Portland, ON, USA, 1990.

Andersson, J. "Multi-Objective Optimization in Engineering Design". PhD thesis, Linköping University, Linköping, Sweden, 2001.

Blackenfelt, M. “Managing Complexity by Product Modularisation - Balancing the Aspects of Technology and Business During the Design Process”. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2001.

Clarkson, P. J., Simons, C., Eckert, C. “Predicting Change Propagation in Complex Design”. Proceedings ASME DETC-2001, Paper No DETC2001/DTM-21698, Pittsburgh, PA, USA, 2001.

Clausing, D. “ Total Quality Development”. ASME Press, New York, 1994.

Dahmus, J. B., Gonzalez-Zugasti, J. P., Otto, K. N. “Modular Product Architecture”. Proceedings ASME DETC-2000, Paper No DETC2001/DTM-14565, Baltimore, MD, USA, 2000.

Erixon, G. ”Modular Function Deployment - A Method for Product Modularisation”. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 1998.

Eppinger, S. D., Salminen, V. “Patterns of Product Development Interactions”. Proceedings of ICED-2001, Glasgow, Scotland, 2001.

Johannesson H.L., Söderberg R.“Structure and Matrix Models for Tolerance Analysis from Configuration to Detail Design”. Research in Engineering Design, 12:112-125, 2001.

Liedholm, U. “On Conceptual Design of Complex Products”. LicEng thesis. Linköping University, Linköping, Sweden, 1999.

Malmqvist, J., Svensson, D. “A Design Theory Based Approach Towards Including QFD Data in Product Models”. Proceedings ASME DETC-1999, Paper No DETC99/DTM-8769, Las Vegas, NA, USA, 1999.

Malmström, J., Malmqvist, J. Trade Off Analysis in Product Structures - A Case Study at Celsius Aerotech. Proceedings NordDesign-98, pp 187-196, Stockholm, Sweden, 1998.

Mortensen, N. H., Andreasen, M. M. “Designing in Interplay with a Product Model - Explained by Design Units”, Proceedings Tools and Methods for Concurrent Engineering, Budapest, Hungary, 1996.

Olsson, F. K. G. “Systematisk konstruktion (Systematic Design)”. PhD thesis, Lund University, Lund, Sweden, 1976.

Pimmler, T. U..& Eppinger, S. D. “Integration Analysis of Product Decompositions”. Working Paper, Sloan School of Management, MIT, 1994.

Schlüter, F. “On the Integration of Environmental Aspects into Early Product Development - Life Cycle Design Structure Matrix”. LicEng thesis, Royal Institute of Technology, Stockholm, Sweden, 2001.

Smith, R. P., Eppinger, S. D. “Identifying Controlling Features of Engineering Design Iteration”. Management Science, Vol. 43, No. 3: 276-293, 1997.

Sosa. M. E., Eppinger, S. D., Rowles, C. M. “Designing Modular and Integrative Systems”. Proceedings of ASME DETC-2000, Paper No DETC/DTM-14571, Baltimore, MD, USA, 2000.

Sudjianto, A., Otto, K. “Modularization to Support Multiple Brand Platforms”. Proceedings ASME DETC-2001, Paper No DETC2001/DTM-21695, Pittsburgh, PA, USA, 2001.

Svensson, D., Malmström, J., Pikosz, P., Malmqvist, J. “A Framework for Modelling and Analysis of Engineering Information Systems”. Proceedings ASME DETC-1999, Paper No DETC99/CIE-9006, Las Vegas, NA, USA, 1999.

Stake, R. “On Conceptual Development of Modular Products”. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2000.

Steward, D. V. “The Design Structure System: A Method for Managing the Design of Complex Systems”. IEEE Transactions on Engineering Management, Vol. EM-28, No 3:71-74, 19981.

Suh, N. P. “Principles of Design”, Oxford University Press, Cambridge, UK, 1990.

Ulrich, K. T., Eppinger, S. D. "Product Design & Development”, 2nd Ed. McGraw-Hill, New York, 2000.

Vallhagen, J. “An Axiomatic Approach to Integrated Product and Process Development”, PhD thesis, Chalmers University of Technology, Göteborg, Sweden, 1996.

Professor Johan Malmqvist
Chalmers University of Technology
Engineering and Industrial Design
SE-412 96 GÖTEBORG, SWEDEN
Phone: +46 31 772 1382
Fax: +46 31 772 1375
E-mail: johan.malmqvist@me.chalmers.se