INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

A VISION TO OVERCOME "CHAOTIC" DESIGN FOR X PROCESSES IN EARLY PHASES

Udo Lindemann
Technical University of Munich

ABSTRACT
There is ample literature available dealing with specific questions regarding Design for X, where "X" takes various meanings such as assembly, manufacturing, cost, and recycling amongst many others. The multidimensional problem spread caused by the different "X" possibilities, generates tremendous complexity due to the large number of interdependencies. In practice, today's product solutions are more or less based on the experience of a few designers and the results achieved are commonly sub-optimal. Developers and researchers increasingly work more in dealing with interdependencies between domains - a specific domain is a situation field that originates from the designer and the design process. Some domains that have a key impact to product development are often overlooked due to the complex nature of the varied development tasks.
Following the influence matrix, the Design Structure Matrix (DSM) and the Domain Mapping Matrix (DMM) methods have evolved in this problem area. By combining the latter two methods and complementing them with strength-based graphs, a powerful and novel methodology has been achieved that is already available today. By adding the possibility of weighting interdependencies, it is now possible to take an important step towards crafting a method that supports designers during the entire product development process.

Keywords: Design for X (DfX), Dependency Structure Matrix, Domain Mapping Matrix, Multiple Domain Matrix, Strength-Based Graphs, Quantified Dependencies, DfX - a Network of Dependencies

1 INTRODUCTION
Engineering design has the task to fulfil a large number of requirements during the development of innovative products. Because of limited time constraints, design strategies usually result in the parallelization of phase oriented processes.
Within the early phases the design principle is elaborated by focusing mainly on function and a general modularization of the product. Having done all this, key requirements such as cost, reliability, safety or recycling are already set at least within a certain range.
Methods and supporting information for these requirements mainly address embodiment and detail design as the next phases. Of course, there are numerous of rules of thumb available, but their validity in a specific situation quite often is unclear.

2 "DESIGN FOR X" TODAY
A large number of methods and rules (rules of thumb) have been developed to support engineering designers fulfil requirements regarding a large number of different aspects. In literature these rules are quite often addressed as Design for X (DfX), where X may stand for assembly, manufacturing, recycling, cost etc. Table 1 may give an impression of the number of aspects which have been addressed in literature; the list is not complete at all. Most of the content of the DfX addresses embodiment or detail design, which is towards the final phases of the engineering design process.
During product concept design, knowledge extracted from personal experience and research results hold that once a principle solution is determined, it has a major impact on defining many of the ultimate product's characteristics. As a result, quite often we try to strengthen the early phases of product development, which in some companies is called "front loading". One example may be cost estimation of a new principle solution, where engineers try to identify characteristics, which may help to identify future efforts in production etc. as a basis for a rough calculation.

Following this point, more methods and more information are required in the early phases of product development. The awareness of the situation in total must be improved and the situation with all the important dependencies has to be modelled.

Table 1. Examples for the "X" of Design for X

accessibility	customs	life cycle costs	production	storage
aesthetics	delivery time	local culture	qualification of staff	sub-suppliers
assembly	disassembly	logistics	quality	synergies
automation	distribution	longevity	recycling	testing
availability	emissions	maintainability	refurbishing	tolerances
build ability	energy consumption	manufacturing	reliability	upgradeability
capability	environment	misuse	repair	usability
cleaning	ergonomics	monitoring	resources	variants
comfort	experience	noise	risk	vibration
company targets	fault tolerance	operability	robustness	wear
competitors	globalisation	operating costs	safety	weight
complexity	innovation	operating errors	security	…
corporate identity	just in time	operation	service	…
cost	knowledge	overload	site flexibility	…
customer	laws and regulations	power	size	…

In the following discussion, some quotations from Ulrich and Eppinger [1] will be used to discuss important conclusions in the context of this paper.

> "For products made in quantities of less than several hundred thousand units per year, this assembly is almost always performed manually. One exception to this generalization is the assembly of electronic circuit boards, which is now almost always done automatically, even at relatively low volumes. There will likely be more exceptions in the coming years, as flexible, precision automation becomes more common." ([1] page 245).

What can we conclude by reading this? First, we realize that there are many examples of products made with automatic assembly with a production volume much less than 100000 per year. Additionally, many manually assembled products exit with a production volume of more than a million per year. That means that the optimal solution depends on the situation, which may be described by a set of characteristics such as the people involved, company, country, competition etc. Second, there is a hint that conditions are changing over time.

> "Assembly labour can cost from less than $1 per hour in low-wage countries to more than $40 per hour in some industrialized nations." ([1] page 246).

This sentence supports the conclusion that the situation at hand has a large impact.

> "DFA index = (Theoretical minimum number of parts) x (3 seconds) / Estimated total assembly time" and "The "3 seconds" in the numerator reflects the theoretical minimum time required to handle and insert a part that is perfectly suited for assembly." ([1] page 251).

Again, the question regarding the situation has to be addressed. Depending on characteristics like size, weight, shape, and material used there may be a required time of approximately 3 second, but it may in fact take on a range of minutes or even hours given a different situation. In addition, one has to recognize that assembly also may include numerous side processes like adjustment, measurement, etc.

> "Integrated parts are often less expensive to fabricate than are the separate parts they replace." ([1] page 252)

and a few lines later

> "Note, however, that part integration is not always a wise strategy and may be in conflict with other sound approaches to minimizing costs." ([1] page 253).

Here again, the statement is dependent on the specific situation and the word "often" is also addressing the heuristic character found in a rule of thumb.

> "Consider the Impact of DFM Decisions on Other Factors" – "Development time" and "cost", "product quality", "external factors" like (e.g. component reuse, life cycle cost). ([1] page 256).

This is an important hint concerning dependencies between the different "X" which have to be kept in mind.

Similar points are addressed by other authors too; here are some examples from Ehrlenspiel [2] and a comment to Otto&Wood [3].

“Accordingly, the costs are dependent in the first place on the following variables:

- Parts count and their joining properties dependent on geometry, surface and material
- Number of subassemblies and their joining properties at the interfaces to other subassemblies or parts
- Connection processes

….

Hard-to-handle parts cause technical problems and thus increase costs; they are characterised by following properties:

- Extreme weight (mass)
- Extreme dimensions and/or size differences
- Coarse tolerances
- Confusing shapes (for example springs, clips, retaining rings, cables)
- High sensitivity
- Extreme physical or chemical properties

…” ([2] page 254).

Again, we find hints for a number of interdependencies between product characteristics and assembly effort that lead to higher cost.

Otto and Wood [3] present a number of guidelines that usually fail to hint under which situation or conditions these specific guidelines are valid.

Table 2 gives an overview of the key conclusions.

Table 2. Key conclusions concerning DfX-processes

1	DfX-processes should depend on the situation
2	DfX-processes should be flexible and have to adapted to changing situations
3	DfX-processes should be aware of the interdependencies within specific domains
4	DfX-processes should be aware of the dependencies between elements in different domains
5	DfX-processes should also support the early development phases

Upon reflection, it becomes clear that a method is required to support engineering design in all phases of product development including documentation, analysis and synthesis.

This method has to handle dependencies in a flexible way with regard to the individual and the specific situation. Experience and knowledge about the specific dependencies within the context of situations described as scenarios or case studies or stories is necessary in support for this method.

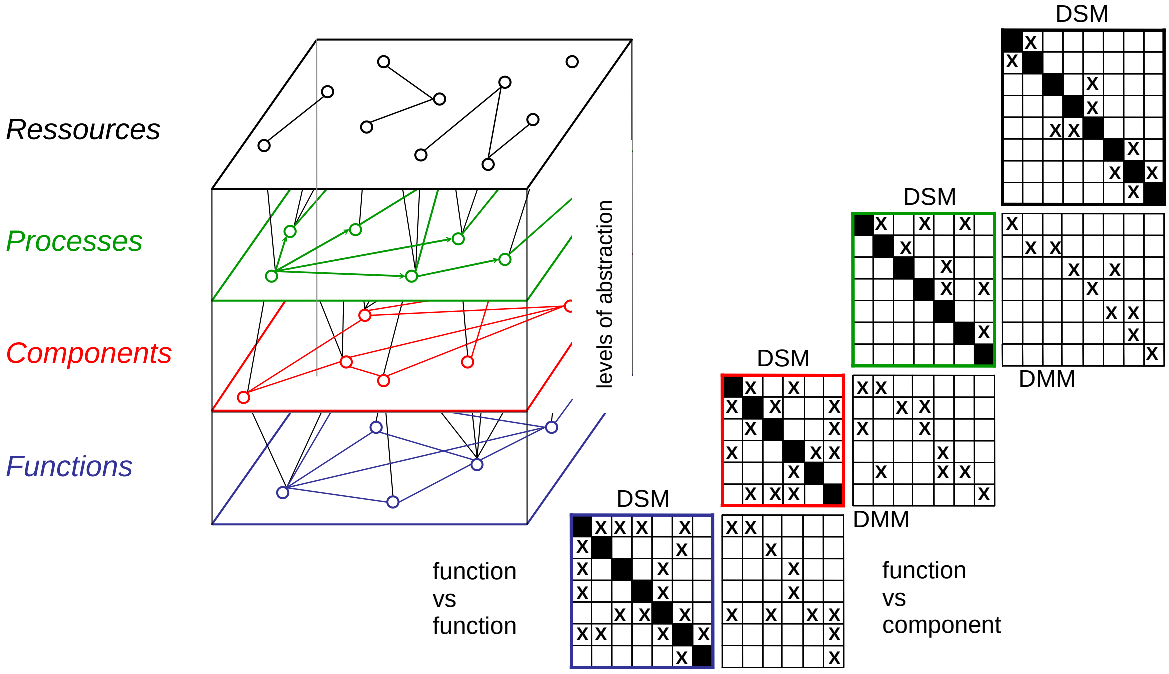
Within the next chapter, a proven method handling dependencies is described briefly.

3 FROM DSM AND DMM MOVING TO MDM

Within the described context, two types of dependency matrices have to be used: the Design Structure Matrices (DSMs) [4, 5] and the Domain Mapping Matrices (DMMs) [6].

DSMs [4] typically focus on interdependencies between elements of one specific domain (e.g., interdependencies between product components or requirements, functions, properties, processes). The information documented by the DSM can also be visualized by strength-based graphs, which is in most situations easier to understand by a designer compared to the matrix because of the specific visualisation. The DMM is a method to visualize the dependencies between two domains, which for example is used within the Quality Function Deployment (QFD) method as the central matrix. In addition, DMMs allow transformation and tracing of information between domains such as functions and components, which may help to verify system models [6]. Although the combination of different domains (e.g., components, functions, and documents) permits formulating important system information, representations for users are confined to interdependencies within the same domain (according to the design structure matrix approach). That allows a better understanding of the system. Thus, a large variety of available algorithms can be applied and visualization is intuitive for users.

Figure 1 shows the dependencies both within and between four different domains. Because of these dependencies, there is a specific DSM for each single domain as for example components, where the dependencies between all components are documented. These DSMs are linked together by a set of DMMs, for example between components and processes.



By combination of several DSMs and the connecting DMMs the Multiple Domain Matrix (MDM) [8, 9] can be generated.
The first task in system definition is to identify the domains which have to be considered for the actual problem, e. g. functions, components, production cost, reliability, maintainability, recycling matters.
Once the concerned domains are identified, interdependency types between these domains have to be determined. For example, the domain "components" can be linked to "function" by the meaning "realizes". Between components there may be a spatial interdependency of importance. Generally, one must distinguish between interdependency meanings of existing data (that will normally be given) and the ones desired for solving the problem.
The interdependency meaning between two domains is shown in the matrix cells. The matrix supports a systematic procedure where all possible domain dependencies are addressed. It may occur that more than one interdependency meaning like spatial and logical is appropriate for connecting two specific domains. For further analysis it is important that these interdependency meanings are stored in separate matrices, because merging will avoid application of most algorithms that are applied later in the process.
At first, one must define the meaning that can be extracted from existing data (e.g., from databases or interviews) [9]. Important criteria for the selection are availability as well as quality of the given data. This selection is case specific and must be done carefully, as further steps are based on this data. Regarding for example the linking of components in figure 2, a relation between two components could represent a spatial link or a general change impact. Interdependency meaning in one domain linking network must be identical for all interdependencies in order to allow later analysis and interpretation. If different linkage types between the same two domains are at hand, related data must not be mixed.

As shown in figure 2, two matrices exist that link between the same two domains (e.g. components and people), differing in inverted link order. If one interdependency meaning between e.g. people and components represents "person works on component", the second interdependency meaning may express "component is processed by person". Thus, the second meaning corresponds to the first one as the passive counterpart. The acquisition of both interdependency meanings can be useful if information is extracted from different sources. In this case, further analysis permits validation by comparing this information. If both mutually related matrices do not emerge from different sources, it is not useful to fill in both, because the opposite matrix can be simply transposed.

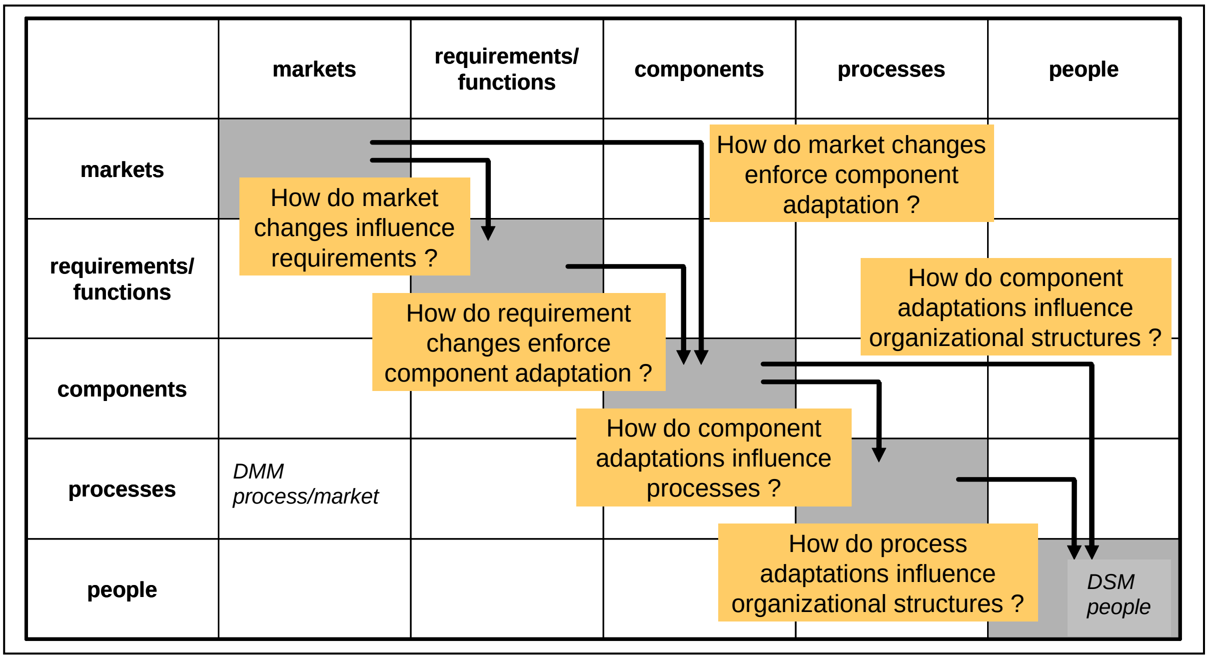


Figure 2. Example of a MDM (Multiple Domain Matrix) [8,9]

The next methodical step is to compute problem relevant intra-domain networks. Depending on case specific structural characteristics (e.g. sub graphs like feedback loops or hierarchies), derived intra-domain networks may either provide useful information or not. For this reason, it seems suitable to set up a computation routine that determines intra-domain networks with low expenditure of time. Consequently, a larger amount of networks can be generated from given domain meaning and verified regarding their significance for further consideration.
Generally, four possibilities exist for computing an intra-domain network from dependencies between two different domains (matrices connecting elements of two different domains are named inter-domain matrices), if only directed dependencies are considered.
A large set of analysis criteria is available for closer consideration of intra-domain networks that were computed by the presented multi-domain approach [9].

The case study observes the student team "TUfast" located at the Technical University of Munich. The team develops a racing-car compliant to the "Formula Student" regulations each year. For the competition, the car design is evaluated from different perspectives: handling, performance, engineering, cost etc.
The TUfast-team relies on a very fast development processes (i.e. time constraints), because each new racecar has to be set up once a year. Although experience from an already realized design process is useful for next year's development, repeated turnover of team members raises difficulties for the group. Against this background, structural information was collected in order to identify critical constellations and provide suggestions for optimization.
The following five domains were identified for the layout of the scenario: components, people, data, process steps, and milestones. The dependencies were directly recorded from the team by interviews and are displayed by grey shading in figure 3. The respective interdependency meaning can be seen in the sub-matrices of figure 3. Altogether, 13 different matrices were available as original data; eleven

are inter-domain and two are intra-domain matrices. Based on this data five intra-domain networks were analyzed in detail. The logic of determination is displayed in figure 3 by arrows. Component and process intra-domain networks were already at hand by original data (round arrows); the other five networks are each computed by two related networks according to the straight arrows. Ultimately, some available matrices were not at all used due to the low quality of the input data (e.g. data-components) or lack of questions / problems of interest.

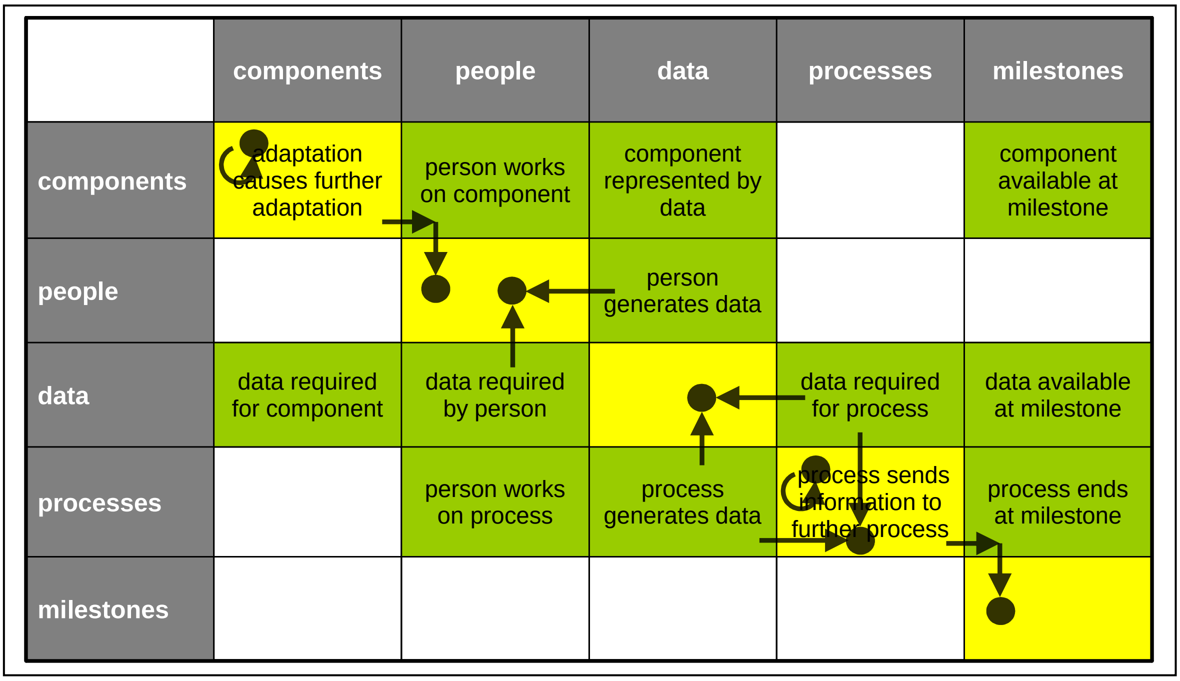


Figure 3. Example of the generation of DSM content [8,9]

At first, the detailed analysis of the component network representing change impact between components is shown. Figure 4 displays the DSM of the components of the TU-fast-racing car in 2006.

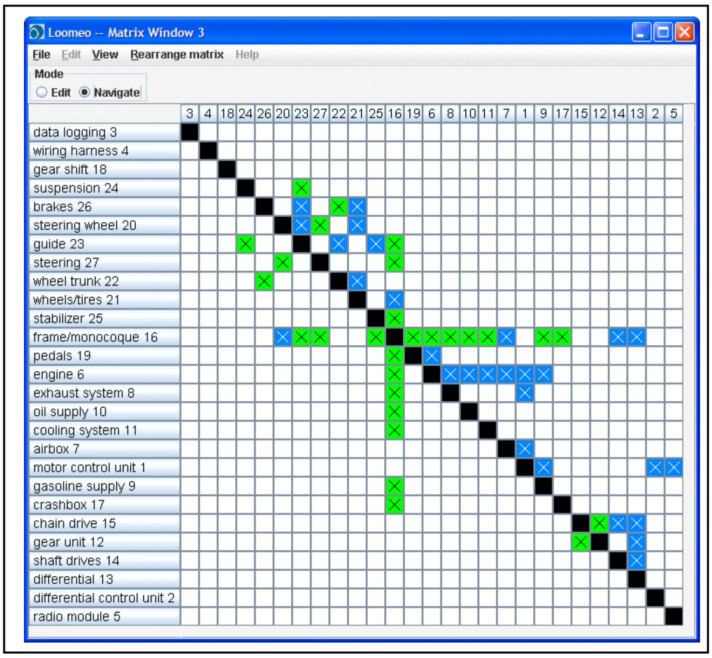


Figure 4. Component DSM [9]

After data acquisition has been finalized the order of elements is recalculated to meet an optimum regarding specific goals. In this case the question was, if there are clearly defined subassemblies. There are two types of dependencies in this matrix, those which are active in only one direction and others in both directions, which are indicated by different colour. There is one central part (frame no. 16) and three further clusters (24-25, 19-17 and 15-14). These representations may give hints concerning modularization of the product and also change effects. Regarding to DfX these clusters may be elaborated with a different degree of interdependencies among each other.

More intuitive for users is the presentation using the identical information by strength-graphs. Graph nodes dispose of repulsive forces to each other and are mutually attracted by graph edges. It is possible to assign weights to graph edges representing further graph information. Edge weights result in different attraction forces (leading to different element distances) between nodes.
As so far all team members take part in the weekly team meeting, figure 5 suggests reconsidering people's communication as well as coordination. Many team members require bidirectional coordination (indicated by double arrows) that cannot perfectly be executed in the overall meeting (otherwise, it is time consuming for all other members). Another point is the change of responsibilities or turnover of team members that can be analyzed on a general point of view.

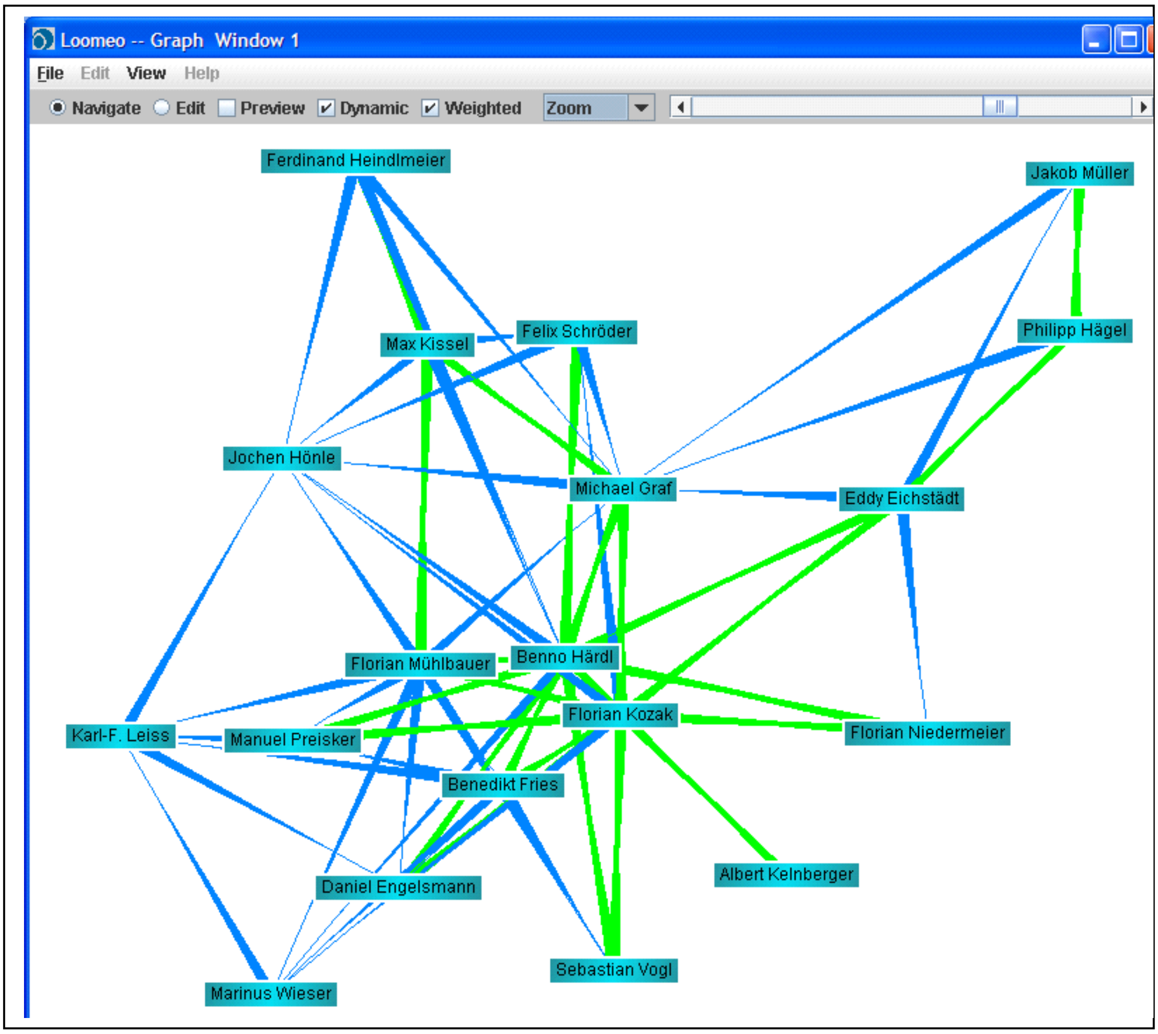


Figure 5. Graph showing interdependencies between people due to responsibility for interlinked components [9]

Figure 6 shows another question of structural relevance in a generic example: these are feedback loops. The illustrated matrix represents a simple situation for example of a process with its sub processes 1 to 8. If the situation is described as seen on the left side in the original matrix it is nearly impossible to find out any critical feedback loop. After reorganising the matrix it is much easier. Looking at some examples from industry a number of some thousand feedback loop have been observed in products with about 25 modules. As it is difficult to see and to get control of the indirect interdependencies this analysis may help to prevent failures when changing the system.

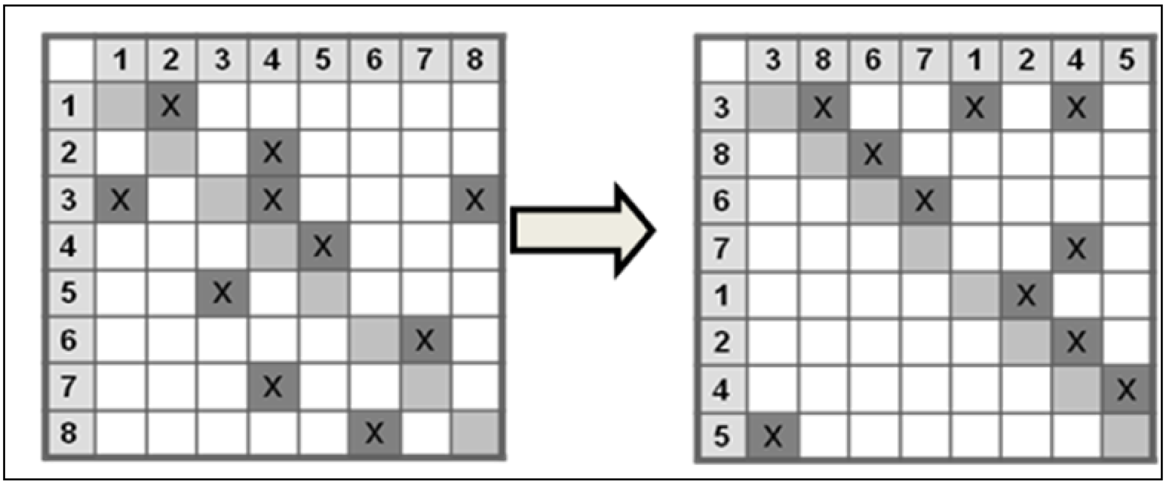


Figure 6. Hidden feedback loops (left) become visible [9]

The use of multi-domain matrices allows for improved data acquisition, because complex interdependency logics can be separated in easier questions more suitable for interviews. Furthermore, the approach of multi-domain matrices provides the possibility of systemic deduction of intra-domain networks, which are suitable for algorithmic analysis as well as intuitive user comprehension. Analysis of intra-domain networks can be executed by means of structure criteria derived from graph theory.
Within another case study a hair dryer was analysed concerning the domains components, parameters, functions, resources and tasks. Fig. 7 shows the result concerning the intensity (low = light red, high = dark red) of indirect interdependencies. A few DfX-topics like noise and vibration and product life are already included in this analysis.

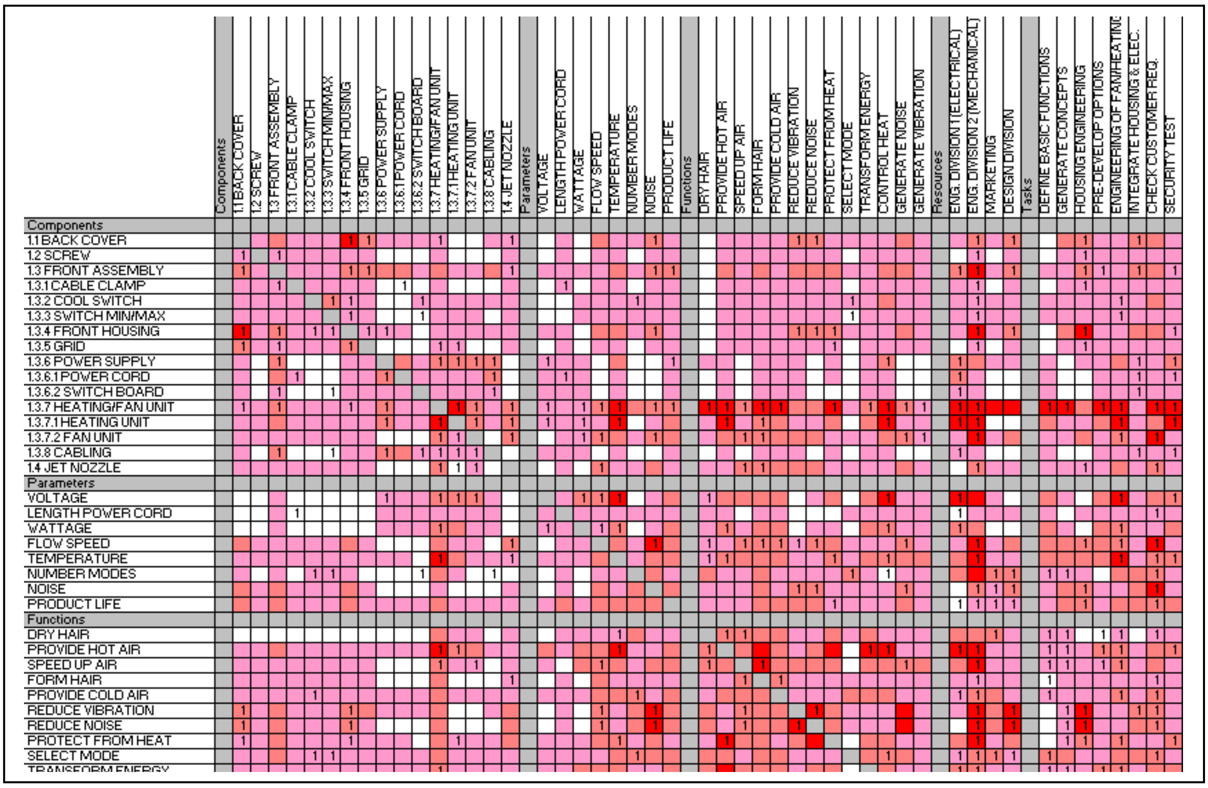


Figure 7. Intensity of indirect interdependencies within a system hairdryer (part of it) [9]

As it is difficult to see and to get control of the indirect interdependencies this analysis may help to prevent failures when changing the system.

The method MDM was proven by a number of case studies in industry with matrices of up to more than thousand elements and about 20 different possibilities to analyse matrices.

4 MDM AND WEIGHTED DEPENDENCIES

The method MDM roughly described the yes/no interdependency between elements. Expanding on this idea, the possibility to work with specific values within the matrices should be considered.
The following description is based on the cost estimation of mechatronic products including their development and testing efforts. The hypothesis is that the development costs are strongly related to interfaces and complex interdependencies.
Because of that, functions, components (mechanic, electric, software) fulfilling the function, processes realising the components and resources used within the processes will be analysed and documented by a MDM. Within and between these domains the entire system complexity can be modelled.
The matrices will be built up step by step starting with functions based on requirements. Within the next step, the system has to be partitioned concerning the mechanic, electric and software domains. Based on this result suitable effects are then selected to bridge components within the three different domains. The structure created with the components above is completed by its interdependencies and as a specific domain the relations between components. Based on this information the processes to develop, test and produce the product are refocused. The objects and their interdependencies within these processes become the process building blocks. Finally the consumption of resources is addressed.
To be able to estimate the efforts and costs the interdependencies have to be weighted for example as weak, medium or strong as long as there are no quantities available. If interdependencies can be quantified, then this should be integrated in the model. The weighting is done by attributes added to the interdependencies.

As an example figure 8 is addressing the emergence of the process-oriented MDM. For all components and on the same level all relations between components the question of which processes will be required to create it is asked and recorded as process building blocks to the process-DSM as part of the whole MDM. The process building blocks are differentiated in those related to components and others related to relations between components (interfaces). The possibilities to analyse the MDM already at this level provides hints concerning specific efforts.

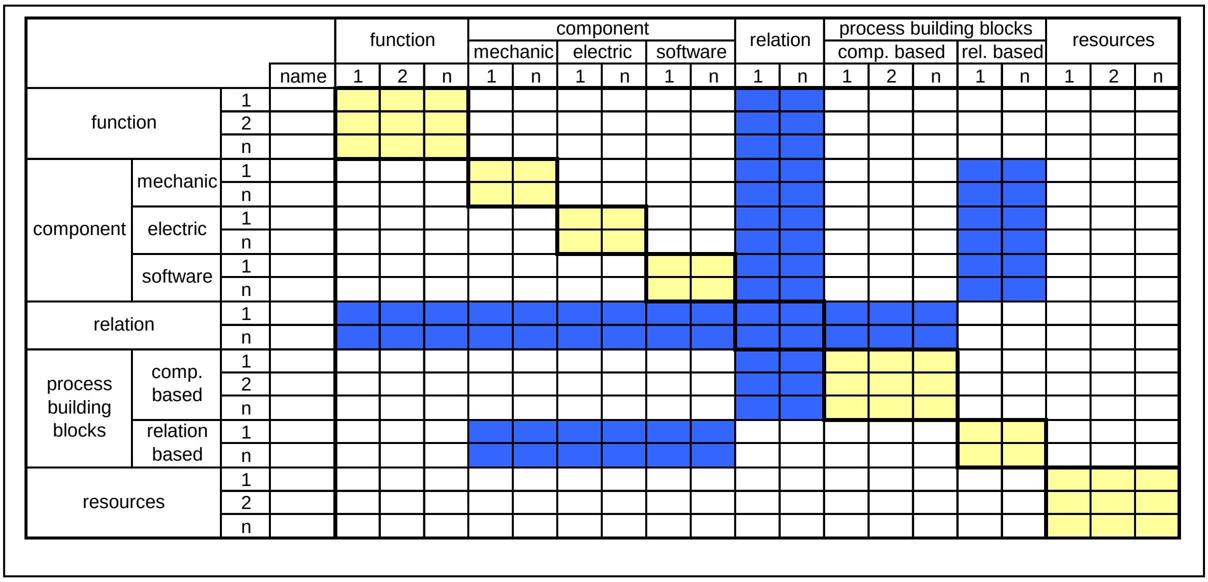


Figure 8. Process-MDM of a mechatronic product (part of it) [7]

To read figure 8 one might go through step by step. A specific function requires one or more components and in addition the relation between components (interfaces). To generate components as well as the above relations processes are required which may be component oriented (mill a part on a

milling machine) or relation oriented (assemble components). Within the final step the resources (staff, material, finance, test equipment, etc.) required to fill the processes will be addressed.
Addressing efforts (costs) it is important to understand, how the DMM linking the process-DSM and the resource-DSM is defined concerning the kind and intensity of the resource consumption. Here the quantification takes place in this example. The intensity of these relations may be modelled by a mathematical formula using a spreadsheet.
This example demonstrates at least in principle how the weighted interdependencies can be modelled.
The research project behind these ideas is an ongoing project funded by the German Research Foundation (DFG).
Funded by an industry partner there is another ongoing project addressing the full range of DfX topics as some kind of Knowledge Management for a specific sub-assembly in the field of gas turbines.

The approach in general has already been tested in case studies; further development is an ongoing process. Compared to classical attempts of optimization tools early phases with insecure and incomplete information are addressed. Key optimization steps are possible concerning the structural complexity of future systems based also on qualitative data.

5 A VISION ABOUT HANDLING "DFX'S"

In the chapters above, a possibility has been shown to address DfX during the early phases of product development. The method described includes a lot of efforts in generating or assembling all the required information. This of course might be a disadvantage which has to be recognised. But on the other hand acting without the addressed knowledge is leading to problems, testing efforts and at least a lot of iterations.
Initially, the interdependency between the design to be undertaken and the given situation was addressed. Because of the importance of this matter the definition of typical scenarios is suggested. Points like recentness, quantities, supplier structure, level of labour cost etc are examples of important characteristics of the scenario. These scenarios have to be defined, collected, compared, judged and documented.
Initially, a set of questions of interest has to be defined in order to keep the effort of defining the interdependencies in the actual situation as low as possible. Following these questions the required input data can be defined. Now the question is how define the required input data without expending too much effort. Regarding these points typical patterns of interdependencies based on specific scenarios should be available in libraries (company specific or within branches). Using and adapting these patterns should reduce the overall effort required in implementing this methodical approach.
Open are questions concerning the interdependencies between the elements (nodes like components, functions, processes etc.) and the dependencies itself (interfaces, dependencies between elements within one domain, dependencies between different domains etc.). Modelling techniques have to be tested and proven at least for a limited set of scenarios. Researchers are asked not only to deepen the knowledge in specific fields (DfX-domians) but also on a system basis.

REFERENCES

[1] Ulrich K. T. and Eppinger St. D. Product Design and Development. McGrawHill Boston 2000
[2] Ehrlenspiel K. and Kiewert A. and Lindemann U. and Mahendra S. Hundal as Editor Cost-Efficient Design Springer Berlin 2007
[3] Otto K. and Wood K. Product Design. Prentice Hall New Jersey 2001
[4] Browning T. Applying the Design Structure Matrix to System Decomposition and Integration Problems: a Review and New Directions. IEEE Transactions on Engineering Management, 48(3), 2001, pp 292-306
[5] Lindemann U. and Maurer M. and Kreimeyer M. Intelligent strategies for structuring products. In: Clarkson J. and Huhtala M (Eds.). Engineering Design - Theory and practice. Cambridge, UK: Engineering Design Centre 2005
[6] Danilovic M. and Browning T. A Formal Approach for Domain Mapping Matrices (DMM) to

Complement Design Structure Matrices (DSM). The 6th Design Structure Matrix (DSM) International Workshop, Proceedings, 12.-14.9.2004, University of Cambridge, Cambridge Engineering Design Centre, Cambridge, UK.
[7] Braun St. and Lindemann U. "Multiplanare Vernetzungen – Abhängigkeiten zwischen Produktkonzept, Produkterstellungsprozess und Ressourcenverbrauch in der Mechatronik. 5. Paderborner Workshop „Entwurf mechatronischer Systeme", HNI Paderborn 2007
[8] Eichinger M., Maurer M, Pulm U. and Lindemann U. Extending Design Structure Matrices and Domain Mapping Matrices by Multiple Design Structure Matrices. In Proceeding of the 8th Biennial Conference on Engineering Systems Design and Analysis (ASME-ESDA06). Torino, Italy, July 4-7 2006
[9] Lindemann U. Maurer M. Facing multi-domain complexity in product development. CIRP and Berliner Kreis: The Future of Product Development. Berlin 2007.

Contact: Udo Lindemann
Technical University of Munich
Product Development
Boltzmannstrasse 15
85748 Garching
Germany
Phone +49 89 289 15130
Fax +49 89 289 15144
e-mail udo.lindemann@tum.de
URL www.pe.mw.tum.de