INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

INTERRELATING AND PRIORITISING REQUIREMENTS ON MULTIPLE HIERARCHY LEVELS

K. G. M. Eben, C. Daniilidis and U. Lindemann

Keywords: requirements elicitation, requirements prioritisation, MDM

1. Introduction

Successful products are a prerequisite for a company's success. Due to fast changing market demands, growing customer orientation and rapidly evolving technologies the complexity of products increases continuously. In order to handle this complexity, it is unavoidable to bestow great care on identifying and analysing requirements demanded on the products. Hence, the latter constitute a basis for development of new products of particular importance in the early phases of the design process. However, the requirements additionally support the assessment of solutions in later phases, whereby the former function as control instrument whether the customer and market demands can be met by the products. Besides these quality issues, a thorough requirements management allows for the reduction of costs by avoiding unnecessary over-engineering.

Further, complexity arises with the involvement of various stakeholders in the development process, since there are not only the customers as sources for requirements but also multiple perspectives on the product can be found within a company. In order to allow for the expedient management of requirements the links between these perspectives have to be known, as are the links between the specific requirements of each perspective.

Especially in small- and middle-sized enterprises (SME) the fast changing markets and customer wishes entail additional challenges, as the allocation of resources in SMEs is often difficult due to the companies' size [Braun, 2005]. Thus, efficiency in the development process is crucial. The same applies for closeness to the customers' problems, in order to enable the creation of successful and innovative products. Hence, the use of systematic methods is seen as necessary [Braun 2005]. Time pressure in every day's work is often the cause for not applying methods in SMEs. Moreover, this is caused by the opinion of engineers in SMEs, who often doubt the methods' practical relevance and assume a high complexity and academic characteristics of the methods; only because they mainly have been developed in research institutes [Braun 2005].

In summary, it is crucial to conduct the management of requirements carefully. Particularly in the environment of SMEs practical methodologies have to be available, which support this task efficiently without unnecessarily committing resources.

2. Handling requirements – State of the Art

Up to now, much research has been done concerning requirements management. Approaches tailored specifically to the field of product development (e.g. [Baumberger and Lindemann, 2006]) can be found in literature. Moreover, as requirements engineering has become a central task in software engineering [Pohl, 2008], the methodological support of handling requirements is of major interest.

The procedures of both environments mentioned above comprise similar steps. E.g. [Maier, et al., 2007] compare various requirements management approaches regarding the following steps. First, there are the elicitation – as means to gather an initial set of requirements – and the analysis of requirements in order to evaluate, decompose and structure the latter. Further, they name the step allocation, which serves to assign requirements to detailed processes or elements of products. The traceability aims at a continuous analysis during the whole development process with the purpose of retracing the source of requirements. The requirements are verified, i.e. tested whether they have been met. Moreover, the requirements are classified hierarchically and their impact on the system is propagated.

The documentation of requirements is often conducted using modelling techniques. Besides methods like for example semantic nets, object oriented modelling – or programming – languages like the unified modelling language (UML) are mentioned by several authors (e.g. [Pohl, 2008], [Maier, et al., 2007]). Moreover, [Kusiak and Wang, 1995] introduce a network model to integrate multiple perspectives on the design task – each bringing along a specific set of requirements. The software-based model is then used to compute multiple design solutions. [Garfield and Loucopoulos, 2009] propose a conceptual framework for modelling requirements consisting, which consists of two units. On the one hand there is the requirements definition – regarding strategic requirement – and on the other hand the requirements elaboration. The latter's purpose is to generate an understanding of the impact each requirement. It uses ontology models, scenario models and a system dynamics model, containing a customer, a financial, a business process and a learning and growth sub-model.

2.1 Requirements and matrix-based approaches

Quality Function Deployment (QFD) is an extensive approach to consider in the product development process. Customer-oriented decisions and product quality can be enhanced during the development process and unwanted over-engineering of product properties can be avoided. Therefore, customer requirements are opposed to the elemental technical requirements in a matrix [Akao 1992]. QFD can be used to decompose abstract customer requests to a product into elemental technical requirements, creating a two level hierarchical requirements structure.

Generally, matrix-based methods are applied to model the interrelations between elements of a system, within a domain – e.g. using a *Design Structure Matrix* (DSM) – and between two domains – by the means of *Design Mapping Matrices* (DMM) [Browning 2001]. In more detail DSMs and DMMs can be used to handle complexity [Lindemann, et al., 2009]. A domain stands for a specific perspective of the system. Thus one step in the application of QFD is building a DMM of the domains customer requests and technical requirements. Various methods, e.g. a clustering or triangularisation of the matrices, can be applied to these structural models in order to gain insight into the systems characteristics or attain hints of how to optimize the system [Lindemann, et al., 2009]. Beyond the mapping of one or two domains, several views are integrated in a *Multiple-Domain Matrix* (MDM) [Lindemann et al. 2009]. Akin to DSMs analysis procedures can be conducted on the captured system to result in a better understanding of it. Matrix-based methods have been applied rather successfully in process management and in the design of product architectures [Browning, 2001]. Further, there are some first applications in the field of requirements management, with promising results.

[Baumberger and Lindemann, 2006] use the DSM and DMM for identifying requirements. Starting in the DSM of components the cause for the relations between two components is investigated as it might be representing a requirement. Moreover, it is suggested to use this approach to plan individual adaptation processes by linking requirements to corresponding components of the product structure.

In order to provide all information afforded for engineering innovation, [Maier, et al., 2007] propose a framework for hierarchical requirements modelling. The information is gathered by manually filling several DMMs linking requirements, working principles, components, component parameter values, test measures and tests. By computing additional matrices, the results can be evaluated and different perspectives on the system analysed. Thus, conclusions can be drawn for the development process.

2.2 Intention of the presented approach

Approaches in literature on requirements management usually aim at holistic approaches. They usually afford establishing data management systems and specific software tools. Thus, all these approaches afford instructing users carefully, bringing along considerable effort in training.

As this work focuses on the development process in SMEs, the goal of this paper is to support the management of requirements without tying up resources needlessly. Further, it is important to provide methods which are not perceived as overly academic, in order to motivate the stakeholders to use them. On the one hand a complete model of requirements is to be obtained, reflecting all levels of abstraction, which yield in requirements. On the other hand the identification and prioritization of requirements is to be supported with minimal effort.

With the purpose of tackling the problems stated above, matrix based approaches are used for the development of a procedure to support requirements management.

As mentioned before, matrix-based approaches already have been used in requirements management with success. But so far, a maximum of two hierarchy levels of requirements are modelled within one view, e.g. a DMM depicting customer requirements and technical properties of a product.

In order to gain a complete understanding of how requirements arise from different levels of hierarchy or various organisational levels, it cannot be sufficient to depict only two layers at one time.

In the case of a complex product system, this system consists of sub units, combined by several subassemblies or modules, which of course comprise single components. On each level different requirements can be found. Hence a model of only two levels of abstraction would result in a rather limited view. Thus, a complete requirements model would be an expedient support during the elicitation, which points out the system's hierarchy levels.

Similar to approaches like QFD [Akao, 1992] or the identification of requirements introduced by [Baumberger and Lindemann, 2006] or [Maier, et al., 2007], the approach presented in section 3 uses the MDM methodology to guide and support the identification of requirements and gain an extensive understanding of dependencies between the requirements over all layers of abstraction. Resulting from building the requirements MDM is one structural model, which can be used to identify critical aspects and conflicts spanning over several domains. All requirements are weighted in order to allow for using them as assessment criteria in the product development process. Thereby, the requirements have to be discussed regarding their validity. Further, it is of great significance that the approach' application provides results instantly. The approach is to enable handling requirements effectively. Whereas, the result must be a detailed understanding of how the requirements are interrelated over all hierarchy levels, in order to estimate the impact of requirement changes.

The procedure presented in this paper is not meant to be used as a global approach for requirements management. Rather, it should serve as a tool for different phases of requirements management. The identification of an initial set of requirements is supported, while all the requirements' levels of hierarchy are documented within one model. Further the requirements can not only be analysed and and decomposed using the matrix-based methods but also prioritized. The completeness of the model is assessed and conflicts between elements are identified. The method can be applied without establishing frameworks of different modelling approaches, software tools and knowledge data bases. Thus, not only the preparation effort for the application of the approach is low. But also, within an enterprise stakeholders' prejudices can be avoided, e.g. against frameworks, which are reputed to be heavily academic. Hence, stakeholders in SMEs may not object to the approach, which can be applied instantly if the need arises to identify and document requirements.

The approach's development and its single steps are described in section 3. Subsequently, it is described how the approach has been applied within a current research project concerning standardisation of pneumatic control systems. Finally, the results are discussed in section 5.

3. Handling requirements on multiple layers of abstraction

Complex products can be regarded from different perspectives on different levels of abstraction, e.g. in pneumatic control systems the highest level of abstraction consists of the control system, which itself is a combination of pneumatic subunits, consisting of several pneumatic devices like valves (for a complete decomposition of such pneumatic control systems see section 4). On each level can be found requirements interrelated in various ways. In order to support the handling of requirements – the elicitation, structuring, prioritising – in this section an approach is introduced using the MDM-methodology. Its focus lies on the support of handling requirements in complex systems of multiple layers of abstraction in small- and middle-sized enterprises. As mentioned above time is a crucial aspect in the resource-limited development process of a SME. Thus the introduced approach aims at demanding a minimum of effort in application, while effects can be seen in short time.

3.1 Development of the approach

The procedure presented below has been developed within a research project carried out in collaboration with a partner in industry. This middle sized enterprise is a manufacturer of pneumatic brake control systems. These systems consist of modules, which are built by combining several pneumatic valves. The aim of the research is to enable the standardisation of the pneumatic valves.
In order to define standard products, for each type of valves the requirements have to be gathered and documented. Since the systems can be decomposed into several levels of hierarchy, it is obvious that each level provides its own requirements.
As within the company requirements are only documented on the system level and afforded changes of valves or single components are communicated by filing design requests, there has been the need for an effective method to support the identification of the requirements for the valves. As resources in the company are limited the procedure using the MDM had been developed. It serves as support to identify and prioritise requirements on the hierarchy levels of valves and components, without changing current processes in the enterprise. Thereby, first the hierarchy levels were defined, represented as the domains in the MDM. After an initial data acquisition first subsets of the requirements MDM had been filled. These subsets have been analysed and prioritised. Moreover structural characteristics, e.g. clusters [Lindemann, et al., 2009], have been identified. These characteristics have been interpreted concerning their meaning in the context of the generated data set. These results have then been used to analyse newly gathered data, supplementing the MDM's first subsets. Thus, by building step by step and employing the results of the analysis to each new data set, the procedure has been adapted and enhanced gradually.

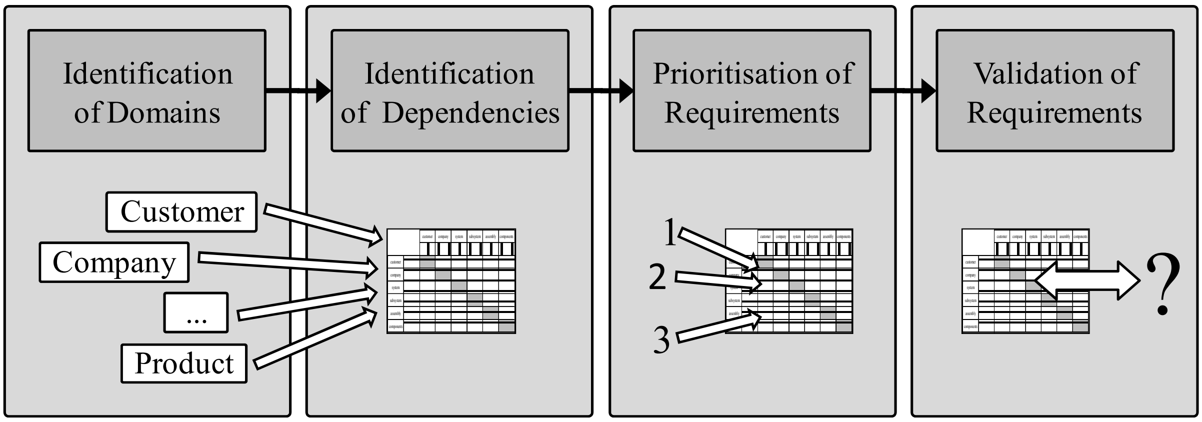

Figure 1. Four steps – prioritisation of requirements using the Multiple-Domain Matrix (MDM)

3.2 Overall Procedure – Using the MDM of requirements

The procedure presented in this subsection comprises of four steps, as it is shown in figure 1. First the product's levels of abstraction on which requirements can be found have to be defined. These levels are different perspectives on the product – different domains – representing various stakeholders in the process and product decomposition layers. Thus the first step consists of identifying the requirements domains. In a second step the requirements of the different perspectives are gathered and their dependencies acquired. Subsequently the requirements are prioritised to serve as evaluation criteria for newly developed solutions later. Finally, the results are discussed and the requirements validated. The first two steps serve the elicitation of requirements. Subsequently the latter are prioritised and in a last step critical points are identified in order to validate the requirements system.

3.3 Step 1 – Identification of requirement domains

There is a large quantity of approaches to support identifying possible sources for and gathering of requirements (e.g. [Pohl 2008]). Often checklists are used to cover all aspects of different views on a product and its surroundings. In this procedure several techniques are used to elicit requirements, as interviews, workshops and analysis of existing documents, e.g. specification sheets, product descriptions or drawings. The gained information is documented using the *Multiple-Domain-Matrix*. The MDM's advantage is its potential to create a common understanding of how the different levels' requirements are linked, i.e. which requirement influences which other requirement. When using MDMs it is possible to regard only certain subsets of a system by filling only one matrix at a time and for instance work on one level of abstraction represented by one domain. The different sub-matrices, DSMs and DMMs, can be combined to form the whole MDM and supplemented by additional matrices, computed from indirect dependencies over several domains [Lindemann, et al., 2009]. So the whole requirements structure can be displayed in one view of the matrix.

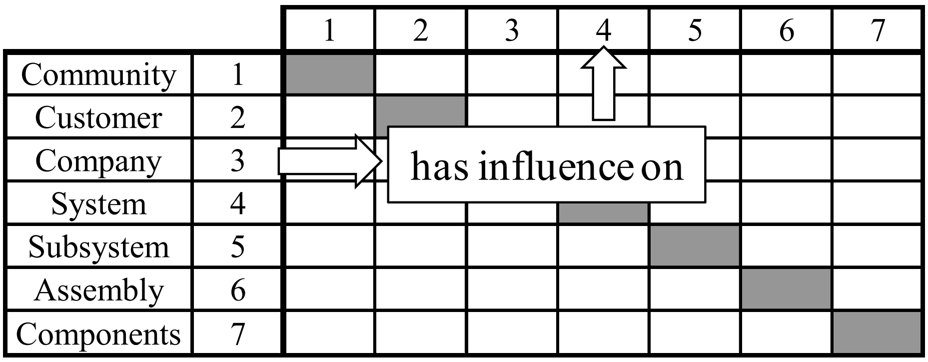

Figure 2. Meta-model of requirements MDM

To begin with, the domains, which are to form the framework for collecting the specific requirements, have to be defined. The domains are one the one hand derived from the hierarchical perspectives on the product of various stakeholders in the development process and on the other hand from product decomposition layers. An important stakeholder is the *community* represented by national and international standards. Further, there are *customers* and the *company's management*, which might for example set target values for maximum production costs or install in house standards. Additionally design teams have to be regarded, which are often divided in sub-teams concerning the product structure. Thus, is important to cover the requirements, which affect the sub-teams' work. This is accomplished by adding the domains *system* (e.g. pneumatic control system), *subsystem* (e.g. section of the pneumatic control system), *assembly* (e.g. valve) and *components* (e.g. seal). Figure 2 shows the seven domains defined above, forming a hierarchical framework for the next steps. The type of relation which is to be regarded in the analysis of requirements is "has influence on" as it serves the purpose of the following analysis steps and can be easily addressed in interviews.

3.4 Step 2 – Identification of requirements and dependencies

After setting up the structure of domains, the requirements comprehended in the domains are gathered. The interrelations of these elements are documented in the sub-matrices, intra-domain, e.g. dependencies between technical requirements of the components, as well as inter-domain relations, e.g. functional requirements of the assembly influencing the technical requirements of the components. Again, this can be attained by analysing existing documents or by conducting interviews. The subsets of the MDM serve as guideline for interviewers, in order to allow an instant classification of attained data and to be able to redirect the interview if the scope of the covered topics is moving out of the regarded system. Moreover, additional subsets can be computed by conducting matrix transformations [Lindemann, et al., 2009; Maier, et al., 2007].

Figure 3 depicts how the links between the elements of the component-requirements-DSM can be derived from the dependencies between the component (CR) and assembly requirements (AR).

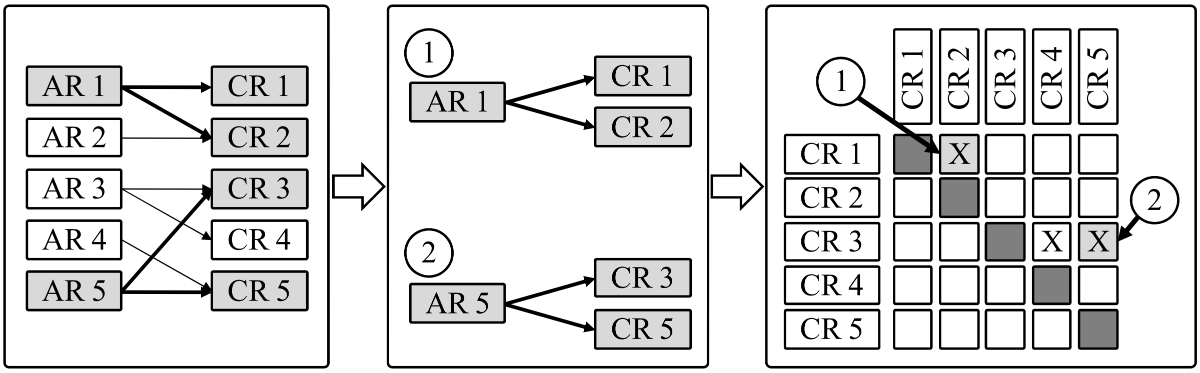

Figure 3. Deduction of component requirements' (CR) dependencies using the indirect links over assembly requirements (AR)

It is likely, that a high quantity of requirements is found and inserted in the matrix. Thus, the necessity arises to structure these aspects within the respective hierarchy levels in order to simplify the subsequent analysis and weighting. This can be done by assigning the requirements to certain groups. In literature a multitude of categories for requirements can be found. For the presented procedure the following categories are chosen to group the requirements within their domains:

- *General conditions*, e.g. international, national standards
- *Strategic requirements*, e.g. target cost, company standards
- *Functional requirements*, e.g. functionalities of products
- *Technical requirements*, e.g. technical properties

Not all categories can be found with all domains. So *general conditions* can only be found within the community, *strategic requirements* mostly with the company management or on the side of the customer, if the product is of importance for his personal or professional success. The domain of components will contain only *technical requirements*, as they usually are the result of requirements on higher levels. Resulting from step two, a thorough understanding of the whole system of requirements is established. The dependencies of the requirements are documented and the origin of lower level requirements can be traced upwards.

3.5 Step 3 – Prioritisation of requirements

According to [Pohl, 2008] requirements can be prioritised based on their importance. He also points to multiple references stating that requirements should be weighted exclusively on one level of abstraction and inherit the importance to lower levels. Thus, mistakes could be avoided, as requirements on higher hierarchy levels implicate higher priorities.

In contrast, here the importance of the requirements in the three domains community, customer and company are prioritised at the same time, although representing three different levels of abstraction. Subsequently, the paths reaching from the requirements in those three domains are tracked down until the component requirements are reached. Each passed element is given the same priority as the starting element of the path, as depicted in figure 4. Hereby, the value 1 stands for the highest priority, 2 stands for a lower priority. The reason for this procedure is that not all requirements could be reached tracing the paths in the matrix and thus not every requirement could be prioritised.

3.6 Step 4 – Validation of the requirements MDM by discussion of results

Finally, it is to discuss whether on the one hand all requirements have been identified and on other hand if the relations have been documented correctly. Structural analysis in the requirements MDM can serve as support to test the plausibility of the gathered data. The considerable significance of this task must not be neglected. As [Biedermann, et al., 2009] states, the importance of correct and complete data acquisition is immense, since e.g. missed or wrongly documented relations can alter the appearance of the regarded system.

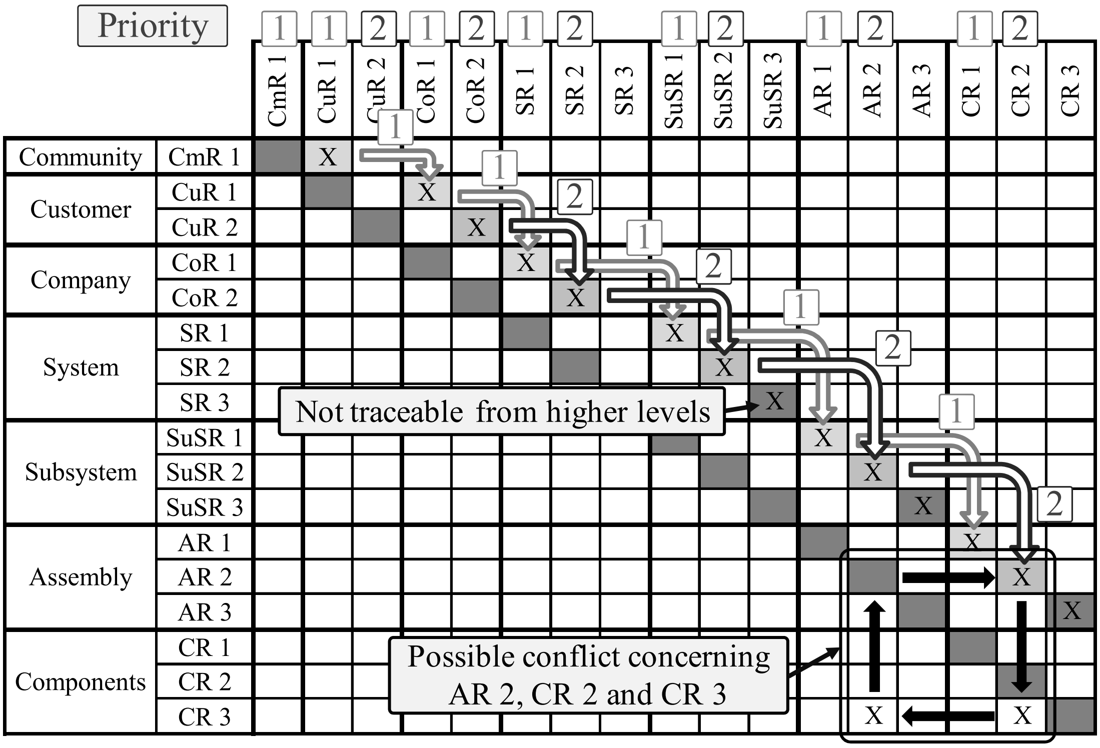

Figure 4. Prioritising of requirements by tracking the paths and identifying critical elements

Several structural characteristics are regarded in order to prove the validity of the gained results or if that is not possible, to identify critical requirements and dependencies. Those then have to be discussed again with the aim of correcting or affirming the current data.

First, the paths of the priorities are examined whether they can be tracked down completely until the domain of the components is reached. If that is not the case it might be that requirements have been missed or dependencies have not been thought of.

Moreover, requirements on lower level domains, not having been assigned a priority value, obviously have no relations to higher level requirements (see figure 4). Here again the question has to be answered, whether relations or elements have been left out inadvertently. If there actually is no connection to the upper levels, this requirement might lack legitimacy. Thus, it could be eliminated after reassuring that no reason for it can be found outside the scope of the MDM.

The weighting's plausibility can be checked by using the MDM to compute the criticality of the requirements in the three highest domains "community", "customer" and "company). The criticality, which is the multiplication of the active sum - the sum of entries in the matrix' rows, the outgoing relations influencing other elements - with the passive sum - the sum of entries in the columns, the ingoing relations [Lindemann, et al., 2009]. If a high criticality results for a requirement, the latter should be given a high weight, the requirement's influence in the system is rather high.

The correct classification of requirements to the four categories general conditions, strategic, functional and technical requirements, can be evaluated by performing a cluster analysis. Here the rows and columns are rearranged until the entries in the cells form highly interconnected clusters around the matrix diagonal [Lindemann, et al., 2009]. The clusters should contain mainly requirements of the same category. Otherwise it has to be questioned whether the category should be changed, or this is just an interface between two categories. For example a functional requirement at some point has to be translated into a technical requirement in order to embody it in the final product, as it is the purpose of QFD [Akao, 1992].

Conflicts between requirements on one level of abstraction can be uncovered by analysing the sub-matrices concerning cycles (see Figure 4). By conducting a search for such closed circuits over two or three domain levels, requirements which influence each other are identified.
Finally, the resulting requirements structure allows for tracking the impacts of requirement changes, by analysing the corresponding paths in the MDM. [Lindemann, et al., 2009] names possibilities to face the propagation of changes.
Summing up, the structural analysis is used to identify requirements which have to be discussed again. As mentioned in section 3 however, there is no guaranty that the collected requirements are complete. Nevertheless the MDM approach serves as support to reach a mature collection of requirements allocated in multiple hierarchy levels.

4. Case Study – Requirements in pneumatic control systems

4.1 Case Study – Applying the Requirements MDM

In order to demonstrate the introduced approach's effectiveness, it is applied to a simplified pneumatic control system of railway brakes. Thereby, it is shown exemplary how requirements and their interrelations are detected across all defined hierarchy levels. Moreover, an example is given of how the prioritisation and validation can be supported.

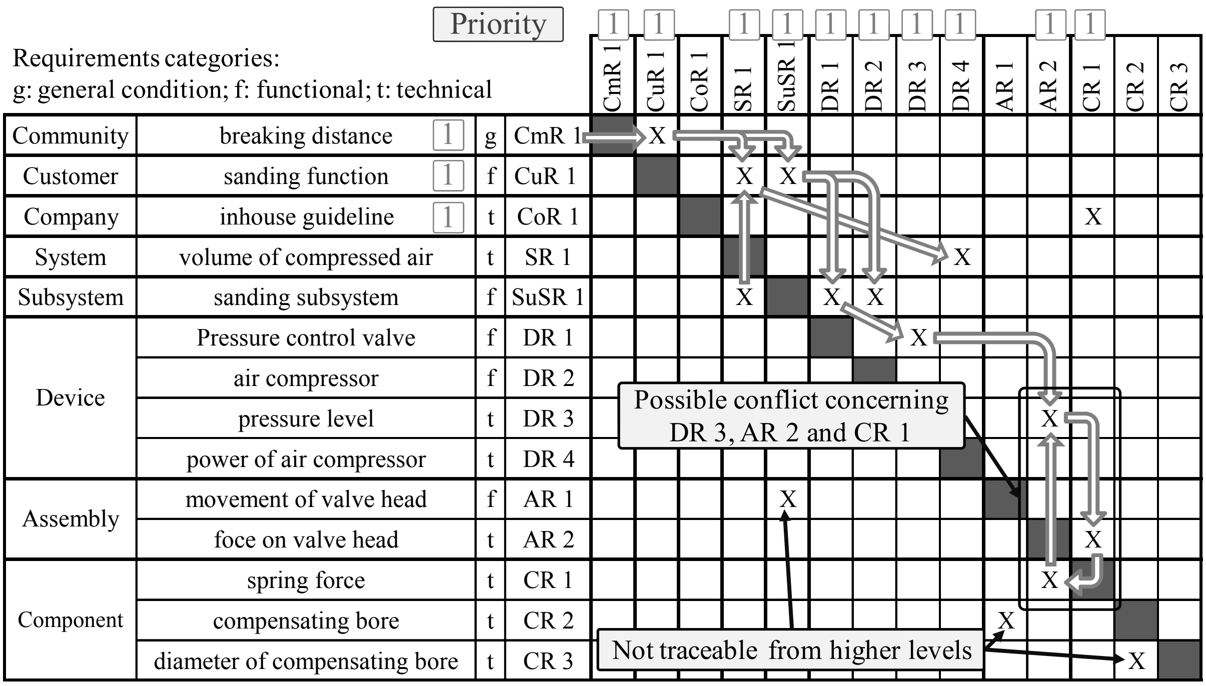

Figure 5. Pneumatic brake control systems – Example for requirements prioritisation

Within the collaborating company, the stakeholders are aware on which of their products' levels of hierarchy requirements exist. However, the dependencies between these levels are not documented and partly only known by certain individuals. Thus, if knowledge of the links between the requirements exists, there is no guarantee that it can be preserved in the future. Hence, it is of great significance to identify the missing information. Moreover, existing and newly gathered data has to be structured and documented carefully. Thereby, the procedure presented in this paper is of great use, since it takes little effort to conduct and is not overly abstract.
In contrast to the hierarchy levels in Figure 2, the control systems' requirements are to be structured in eight levels of hierarchy, as depicted in Figure 5. Besides the domains community, customer, company, the system level stands for the complete pneumatic control systems, the subsystems are combined units, consisting of several pneumatic devices, e.g. valves. The domain, one hierarchy level below, is the assembly domain. For example a valve may comprise a sub assembly of a spring and a

piston, which together have the functionality of moving the piston. Thus, from a functional perspective, they can be seen as a unit with its own functional and technical requirements. Finally there is the domain of component requirements. An example for a customer requirement is the integration *of a sanding function* (CuS 1). This functional requirement is caused by the community requirement *breaking distance*, as the sanding function is used to increase the friction on the rail-tracks, if – e.g. due to snow fall – the tracks' surfaces are slippery. The breaking distance is regulated by national standards and thus belongs to the category general conditions.

The sanding system – subsystem of the control system and caused by the customer requirement sanding function – has an impact on the system level, specifically on the necessary *volume of compressed air*, and the subsystem's configuration itself. The overall necessary volume of compressed air influences the *power* requirement *of the air compressor*. One of the subsystem's pneumatic devices is a *pressure control valve*, which requires a certain *level of pressure*. The latter affects the required force on valve head, which itself results in a certain spring force, also influenced by the strategic requirement *inhouse standard.* As all the requirements mentioned above can be reached over a path starting at *braking distance*, the same priority is assigned to them. The latter can be found the hierarchy level of community requirements; hence it is given the highest priority.

There are three requirements, which cannot be traced upwards to other domains. These can be easily identified as the *movement of the valve head*, the *compensating bore* and *diameter of compensating bore*, as they have not been given a priority. Thus, the question has to be answered, why there is no link to other requirements. In this case the requirements are necessary, as they concern the pressure control valve. In order to control the pressure a valve head is to be moved smoothly when opening and closing the vent. Hence, the relation between the requirements pressure control valve and movement of valve head has been missed. As has the link to *force on valve head*, since the bore allows for equal pressure levels on both sides of the valve head. Thus, the force required to move the valve head is lower. Figure 5 shows a cycle between the requirements *pressure level* (DR 3), *force on valve head* (AR 2) *and spring force* (CR 1). Such closed circuit can serve as an indicator for conflicts between requirements. The underlying factor here is that the pressure level is highly dependent on the design of the spring. Thus this aspect has to be treated carefully when the spring is designed and respectively the pressure level is modified.

4.2 Discussion of Results

By using the approach of the requirements MDM to the pneumatic control system the identification of requirements has been supported. The documented relations between the latter served as basis for the weighting of the single demands. Potentially missed relations have been detected by analysing the paths upwards from lower to higher level requirements and reassessed concerning their validity.

The result is a complete set of prioritised requirements in each of the hierarchical domains. The knowledge of their interrelations not only has been identified but also documented. Moreover, the results allow for the assessment of solutions created in later phases of the development process.

By now the approach has only be used by single project team within the company. Thus, although the results could be gained without considerable effort, it is to clarify how existing ways of requirements documentation – however they are not standardised and often lack completely – could be integrated in the approach, in order to make use of this knowledge.

5. Conclusion and Outlook

In this paper a procedure has been presented using the Multiple-Domain Matrix to support the identification of requirements by structural modelling. It is further used to prioritise the requirements, assigned to seven domains on different levels of hierarchy. The requirements validity is analysed by the use of structural analysis. Thus, not only critical elements and conflicts are identified, but also the impact of a change on other requirements can be estimated. As the approach is meant to be applied within small and middle sizes enterprises, it was developed with the goal not to afford extensive methodological training or installing methodological frameworks, comprising of different databases [Garfield and Loucopoulos, 2009]. It can be conducted efficiently while providing results promptly. The application in a case study showed that the requirements MDM allows for the documentation of

interrelations between the requirements. Moreover, a thorough understanding of the dependencies between the different perspectives on the regarded product can be created.
The proposed procedure should serve as a tool in requirements management. In order to integrate the approach into the development processes permanently, it is recommended to use it in combination with other approaches as proposed for example by [Maier et al. 2007] or [Garfield and Loucopoulos 2009]. The reason is that the matrix is static. Thus, if there are requirements changes the MDM has to be rebuilt, as the changes cannot be modelled yet. Further the structural analysis has to be validated within other projects, in order to ensure the correct interpretation of the identified characteristics in a different context as in the research project. Moreover, it is to be clarified whether the approach is sufficient for the use in larger companies producing more complex products and comprising additional stakeholders on multiple hierarchy levels.
Finally, it has to be taken into account that the approach presented in this paper focuses on the support of certain phases of requirements management. It does not serve as a knowledge base in order to reuse instances of requirements in following projects. The advantage of the MDM supported approach is that it is straight forward and can be applied effectively without tying up resources unnecessarily.
Although there is still work to do in order to achieve a maximum benefit of the presented approach, it already serves the purpose of supporting the identification, weighting and validation of requirements.

References

Akao, Y., "QFD – Quality Function Deployment", Landsberg/Lech: Moderne Industrie, 1992.

Baumberger, C.; Lindemann, U., "Requirement oriented process planning and configuration, Proceedings of NordDesign 2006, Jónsson, M. Þ.; UnnÞórsson, R. (eds.), Reykjavik, 2006, pp. 244-254.

Biedermann, W., Kreimeyer, M., Lindemann, U., "Measurement System to Improve Data Acquisition Workshops", Proceedings 10th International DSM Conference, Kreimeyer, M., Lindemann, U., Danilovic, M. (eds.), Hanser, Munich, 2008, pp. 119-134.

Braun, T. E., "Methodische Unterstützung der strategischen Produktplanung in einem mittelständisch geprägten Umfeld", Dr. Hut, Munich, 2005.

Browning, T. R., "Applying the design structure matrix to system decomposition and integration problems: A review and new direction.", IEEE Transaction on Engineering Management, 48, 3, 2001, pp. 292–306.

Garfield, J., Loucopoulos, P., "Facilitating Requirements Negotiation: Modelling Alternatives and Arguments", Proceedings of the Fifteenth Americas Conference on Information Systems, San Francisco, 2009.

Kusiak, A., Wang, J., "Dependency Analysis in Constraint Negotiation", IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No. 9, 1995, pp. 1301-1313.

Lindemann, U., Maurer, M., Braun, T., "Structural Complexity Management – An Approach for the Field of Product Design.", Springer Berlin, 2009.

Maier, J. R. A., Ezhilan, T., Fadel, G. M., Summers, J. D., Mocko, G., A Hierarchical Requirements Modeling Scheme to Support Engineering Innovation. In International Conference on Engineering Design, ICED '07, Vol. 5, Paris, 2007.

Pohl, K., "Requirements Engineering – Grundlagen, Prinzipien, Techniken", 2nd edition, depunkt.verlag, Heidelberg, 2008

Dipl.-Ing. Katharina Eben
Scientific Assistant
Technische Universitaet Muenchen
Institute of Product Development
Boltzmannstrasse 15
85748 Garching, Germany
Telefone: +49 89 289 15132
Telefax: +49 89 289 15144
Email: katharina.eben@pe.mw.tum.de
URL: http://www.pe.mw.tum.de