ICED13
INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED13
19-22 AUGUST 2013, SUNGKYUNKWAN UNIVERSITY, SEOUL, KOREA

BRINGING OBJECTIVITY TO REQUIREMENTS – USING PROPERTY NETWORKS FOR A MORE COMPLETE REQUIREMENT ACQUISITION

Benjamin RÖDER, Tim GLÄßER, Herbert BIRKHOFER
Technische Universität Darmstadt, Germany

ABSTRACT

The acquisition of requirements is a crucial phase during the product development process. The created list of requirements determines costs, necessary time and iterative steps needed during the product creation. However the level of standardization during the acquisition of requirements is very low due to the vastly different circumstances in every project. This paper elaborates a new systematic approach to acquire a more complete list of requirements based on the use of objective requirement clusters. The use of property networks as basis for the clustering of requirements allows to define clusters in a standardized and highly automated manner. By improving the standardization during the requirement acquisition the amount of unnoticed implicit requirements can be reduced, and incomplete or incorrect requirements can be prevented.

Keywords: requirements, requirement clusters, property networks

Contact:
Benjamin Röder
Technische Universität Darmstadt
Produktentwicklung und Maschinenelemente
Darmstadt
64289
Germany
roeder@pmd.tu-darmstadt.de

1 INTRODUCTION

The acquisition of requirements is one of the most important and also the most complex processes in the development of a product. Whether the various and often complex dependencies between individual requirements can be identified and taken into account during later phases of the development often depends on the expertise and experience of the engineer. This fact, combined with problems that arise due to communications between developers and customers, can lead to an incorrect or incomplete list of requirements. The quality and completeness of the identified requirements vastly influence the costs and the time required for the following processes. A major problem in terms of the completeness of requirements is the field of implicit requirements, since they are not explicitly mentioned by the customer yet are expected to be fulfilled (Pahl and Beitz 2007).Today there are several methods, for example, the main feature lists or the scenario technique (Pahl and Beitz 2007), that are designed to support the developer during the acquisition of requirements. However, they are mostly associative methods and the selection of the right method and its adaptation to a specific project requires a high level of experience in the field of requirement acquisition. To find all necessary requirements using these methods, numerous iterative steps are necessary because requirements are often only detected during later phases of the product development process. As a result, more time and money than necessary is wasted on these iterative steps. The project division product development at TU-Darmstadt's Collaborative Research Center 666: Integral Sheet Metal Design with Higher Order Bifurcations develops algorithm-based methods for the development of products. To reach a higher level of standardization, requirements are acquired using the methodology of requirement clusters (Röder et al. 2011). In the next step, the acquired requirements are combined with property networks to derive target functions and optimization variables for the mathematical optimization of the sheet metal profile. This paper addresses a process that is the opposite of CRC 666's envisaged flow of information, as marked by the circle in Figure 1. It examines the creation of requirement clusters based on property networks and their use for the acquisition of requirements. By using property networks that are based on physical laws, norms or applicable laws, the created requirement clusters are more objective and complete than when using subjective sources. These clusters are part of a requirement cluster catalogue that serves as a knowledge database for creating requirement lists. Using this knowledge database in the process of requirement acquisition ensures that the created list of requirements is more complete. The insights gained from the study of the process of creating requirement clusters will be used in later research to improve the interface between requirement clusters and property networks and increase internal consistency of the processes being executed within CRC 666.

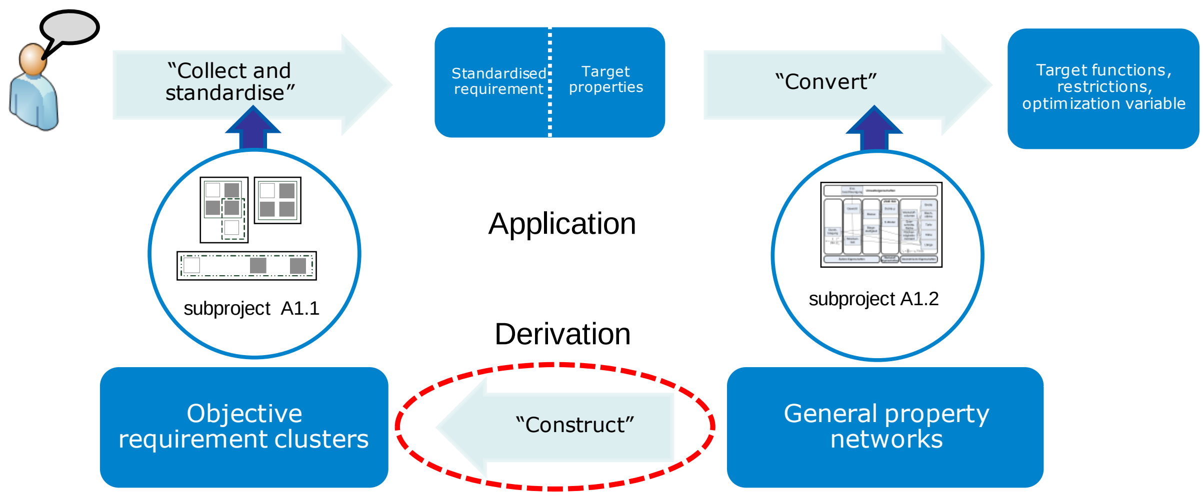

Figure 1. Examined process

Section 2 describes the methodology of requirement clusters defined by Röder et al. (2011). This section discusses the main idea of the approach and the different types of creating clusters and activating them in development projects. The benefits of the approach are also described. Section 3 explains the methodology developed for the transformation of property networks into requirement clusters. The differences between properties of the initial network and the requirements determined by the requirement list as well as the necessary steps to transform them are described. Section 4 is a case study to display the advantages of this method. Section 5 summarizes the results of this paper and gives an overview of the possibilities and limitations of the developed method.

2 METHOD OF REQUIREMENT CLUSTERS

The method of clustering requirements is a known concept, however, the approach described by Röder (Röder 2011) is fundamentally different from previous clustering methods. Clustering methods are used in the field of requirement management, meaning after the acquisition process. They are used to structure identified requirements using a given hierarchy. Therefore, their aim is to improve clarity in subsequent processes. The methodology of requirement clusters by Röder is designed to improve the process of requirement acquisition. Thus, general requirements are clustered and saved in the form of a requirement cluster catalogue that can be used for the requirement acquisition of a specific project. Naming one requirement or a defined term during the process of requirement acquisition can activate other requirements that belong to the same cluster as the initial input, meaning they have some kind of connection. The activated, standardized and project-specific requirements are given to the customer, who can specify a target value for the requirements or add their necessary requirements. The requirement list that is created afterwards consists of the standardized, product-specific requirements and those that have been added by the individual customer. This method tries to improve the acquisition of requirements by identifying relevant requirements by using requirement clusters and reducing the time needed due to the standardized procedure.

An integrated requirement model has been developed to categorize and describe the different kinds of requirement clusters. The different spaces of the model imply different types of requirement clusters so that a complete description and classification of all possible cluster types is possible. Besides the categorization of different requirement cluster types, the model can be used to derive methods for the creation and usage of the different cluster types. Due to the wide range of requirements and requirement sources the integrated model combines the key elements of various other models to cover all requirement types that can occur. The key elements of the integrated model are the product pyramid model in the variation by Sauer (Sauer 2006); the common idea to divide requirements in a solution and a problem space category; approaches for an algorithm-based product development process leading to the systematic restriction of the solution space (for example, Wäldele 2008); the concept of process modeling by Heidemann (Heidemann 2001); and the basic utility theory of economics. These model elements are combined to give an integrated view over the range of possible requirements and their connections to derive the requirement cluster. The complete integrated requirement model can be seen in Figure 2. After a short survey of the model it is used to classify the method described in this paper in the overall context of the requirements cluster methodology.

The overall model space is divided into a solution neutral space, a solution specific space and a restriction space. A basic rule of requirement acquisition states that the customer should define solution-neutral requirements to prevent unintended and unnecessary limitations in the field of possible solutions. However a customer must always have the opportunity to define non-solution-neutral requirements, although they should be scrutinized. Requirement clusters of layers of the solution specific space should primarily be defined by the development team before and during the development process. Some of these clusters are a non-solution-neutral transformation of the solution-neutral requirements, transferring them out of the solution neutral space.

The top layer of the pyramid model describes a Pareto-inefficient condition of the customer. An idea or information triggers a process through which the customer realizes that there is a target condition state (S2) that he prefers to the current condition state (S1) based on his aggregate individual utility preference function (U). At this layer, requirement clusters describe a relative deficit of the customer. The current condition and the preferred target condition can be described using property vectors (Gramlich 2013). The customer has a unique preference (p) for each property vector, which valuates the attributes and different values of a vector. The mathematical description can be seen in the formula below.

Thus property vectors can be ranked and brought to an ordinal order based on the overall utility function. Regarding requirements, the property vectors (E) can be divided into base vectors (BV) and side vectors (SV). Base vectors describe the core target property vectors that cause the Pareto inefficiency and describe the desired target state by which the customer can achieve a higher level of utility. The base vector describes the current and desired state of the operand in the process of usage. If the desired properties of the base vector are not achieved through the development process, the entire development process is unsuccessful. The basis vectors of the current state and desired target state define the solution space of the development project.

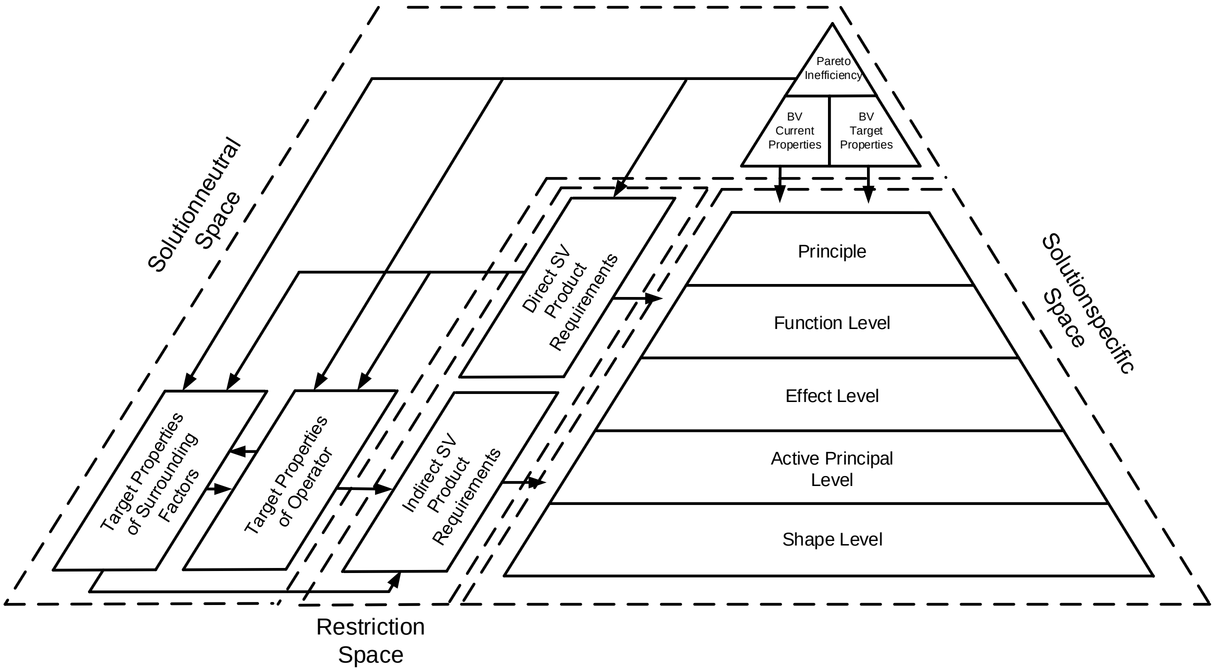

Figure 2. Integrated model for requirement categorization

$$\begin{pmatrix} E_1 \\ E_2 \\ E_3 \end{pmatrix}_{S_1} p_{BV} + \begin{pmatrix} E_4 \\ E_5 \end{pmatrix}_{S_1} p_{SV} = U_{S_1} \leq U_{S_2} = \begin{pmatrix} E_1 \\ E_2 \\ E_3 \end{pmatrix}_{S_2} p_{BV} + \begin{pmatrix} E_4 \\ E_5 \end{pmatrix}_{S_2} p_{SV} \quad (1)$$

The properties of a side vector can also increase the utility level, but only in combination with the transformation of the base vector to the desired state. Side vectors can be divided into two categories. Direct side vectors of product requirements are non-solution-neutral requirements describing the product itself and are directly named. Indirect side vectors of product requirements are non-solution-neutral requirements derived from process requirements following the idea of Schott. On the other hand solution specific requirements can result in solution neutral (process) requirements. (Schott 1997). Requirement Clusters are a helpful tool to support both transformations. Side vectors always limit the solution space.

The core of the solution space is the product model pyramid. From each possible solution at each layer a connected requirement cluster can be derived. These requirement clusters gather the essential properties that define the specific solution. The fixation or ranging of connected values can be linked to it. This implies that these requirement clusters are not solution-neutral. The vector model of target properties is able to describe different phenomena in the field of requirements. Implicit requirements can be described as a set of property vectors for which the customer knows their preference and the properties, but doesn't mention them. Fuzzy requirements are property vectors for which the customer knows the preference and the name for the overall vector, but is not able to name particular properties of the vector. An example of a soft requirement vector would be the requirement cluster "design".

The method described in this paper aims to create requirement clusters that are located in the solution space of the model. It seems logical to orient a requirement cluster catalog of the solution space towards existing solutions. The literature offers a wide collection of catalogued solutions for every layer of the solution space. The descriptions are mostly in the form of formulas and can be transformed into property networks. The creation of property networks is not the focus of this paper and following up the property networks will be seen as an existing base for the method.

3 TRANSFORMATION OF PROPERTY NETWORKS IN REQUIREMENT CLUSTERS

This section deals with the development of a methodology that can be used to structure the connections between product properties illustrated in a property network and, based on this, uses this structure to define requirement clusters. A property network can be used to depict product properties and their relations to other properties. The relations depicted can be of causal or logical nature. Property networks can be determined by analyzing an existing system or physical model or using formulas or material charts and by using informal relations like general construction and design rules (Wäldele 2012). The aim is to use such a network as a basis for defining requirement clusters that can be used to create a complete and correct requirements list in a highly automated manner. The creation of requirement clusters consists of two steps. Section 3.1 deals with the classification of the property network on the basis of its structure using the graph theory. Section 3.2 describes the transformation of the structured property network in requirement clusters.

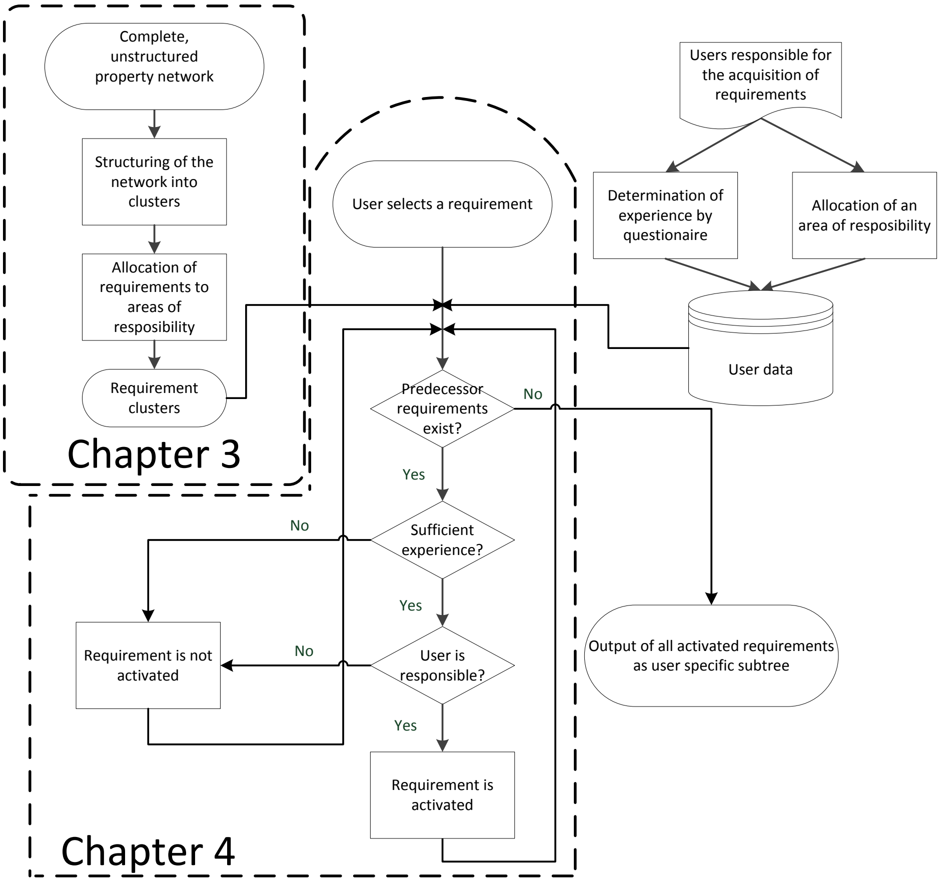

Figure 3. Overview of the envisaged requirement acquisition process

3.1 Structuring of property networks

The aim of this method is to create requirement clusters that can help acquire a more complete and correct requirement list on the basis of an underlying property network. To facilitate this it is necessary that the property network contains every property that was given a target value during requirement acquisition. Therefore, the underlying networks consist of a large number of properties. As a result of the complexity it is very difficult to comprehend or work with the network as a whole. For this reason it is necessary to structure the property network. To facilitate the handling of very large property networks, methods known from the mathematical graph theory were combined to structure the network. The graph theory allows the depiction of a network as an adjacency matrix that can be investigated and structured using relatively simple matrix operations (Turau 2009). The usage of this mathematical description facilitates the structuring of any property network without limitations in the networks. The developed method structures the network based on two characteristics, the first being the distance between dependent and independent properties of the network. In graph theory this

distance is the length of a path between two vertices. The property network can be divided into different levels, depending on the longest path length between a dependent property and the independent properties of this network. These levels can be used to decide whether or not additional requirements are activated for a user, depending on their experience, as described in Section 4.
In addition to this division into levels, every property is assigned to one or more sub-trees, depending on the sinks connected with this property via paths. The sinks of the network are the properties that have no outgoing edges, which means that there are no properties that depend on their property value. This second characteristic allows structuring of a network that can contain information about different contexts into individual areas. These areas completely depict one sink of the network and the properties it depends on and can be examined separately. The individual sub-trees can be used as non-solution-neutral requirement clusters within the solution space, as described in Section 2. This allows the display to the user of only the requirements of the currently relevant sub-tree during the requirement acquisition process. By doing this, the network is disentangled, which allows the user to focus on requirements important to current work. In the following, the mathematical steps to structure the network are expounded with the help of the example property network of a Euler column.
First, every property is assigned with a number (Figure 4). By doing this, it is possible to identify properties that appear more than once in the network. Additionally, it allows the generation of the adjacency matrix of the property network. By using the adjacency matrix of the network, the network can be examined and changed by using matrix operations known from the graph theory (Turau 2009). This promotes the aspired automation of the process.

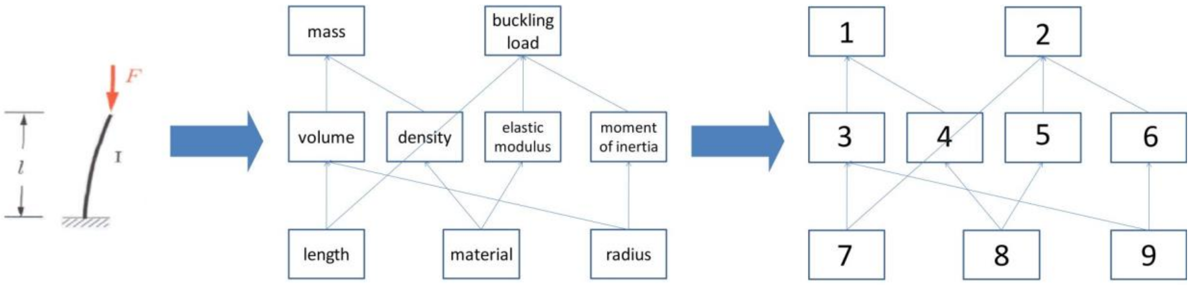

Figure 4. Assigning of distinct numbers to the property network

To divide the property network into levels, the column totals of every column of the adjacency matrix are formed. If a column total equals zero, the respective vertex is a source of the graph, which means that it is independent of other properties. In this case, the value of every entry inside the row belonging to that vertex is set to zero. The value of the entry on the matrix diagonal is set to a count variable that represents the level of the property. After every column total was formed, the count variable is increased by one. This step is repeated until no more column totals equal zero. The result is a diagonal matrix whose values are the level of their respective properties inside the property network. This division into levels is later used to decide whether or not properties are activated for certain users. This can reduce the number of properties displayed simultaneously to the user and therefore simplifies the task of identifying necessary requirements. To structure the property network into sub-trees, it is necessary to generate the adjacency matrix of the transitive closure of the initial network. The sub-graphs of the property network consist of one sink and all vertices that are directly or indirectly connected with the sink via paths. The sub-graphs can be determined by examining the columns of the adjacency matrix of the transitive closure belonging to the sink. The division of the network is done based on the sinks because they normally will be properties of high importance since the network was designed to depict this particular aspect. The division of the network into multiple parts depicting individual aspects of the product allows the user to focus on a single aspect of the product.
In the last step, the initial property network is divided into property clusters with the help of the level diagonal matrix and the matrix of the transitive closure (Figure 5).

3.2 Transformation of properties into requirements

First, the difference between properties of the property network and the later acquired requirements of the requirement list should be discussed. Product requirements are defined in the DIN EN ISO 9000:2005 as an expectation or demand that needs to be fulfilled by the product (DIN 2005). Aside from that definition, it is important for the use of property networks for the acquisition of requirements that the importance of properties differs greatly dependent on the individual user. Instead of defining target values for every property a product developer will focus on those that they personally assume to be important. A property network is a complete and objective depiction of product properties that are difficult to comprehend by the user because of its complexity. For the acquisition of requirements, the user focuses on properties important to the product's success from their point of view. They are normally only small parts of the property network (Figure 6). To help the user with the requirement acquisition only requirements that they comprehend and that are part of their area of responsibility should be activated. Meanwhile, the definition of conflicting requirements should be prevented. The developed method displays additional requirements to the user based on a first manually defined requirement. Whether or not a requirement in the cluster is shown to the user is decided by the experience of the user and their area of responsibility. The experience of the user is determined by questionnaire and saved in a user database. The experience is used to determine the level up to which requirements are activated. This prevents users who are lacking in experience from defining incorrect target values for properties that have a large impact on the product. This is necessary because the impact of one property on the property network as a whole is hard to see. In addition, this reduces the number of simultaneous activated requirements and therefore simplifies the process for users who lack experience.

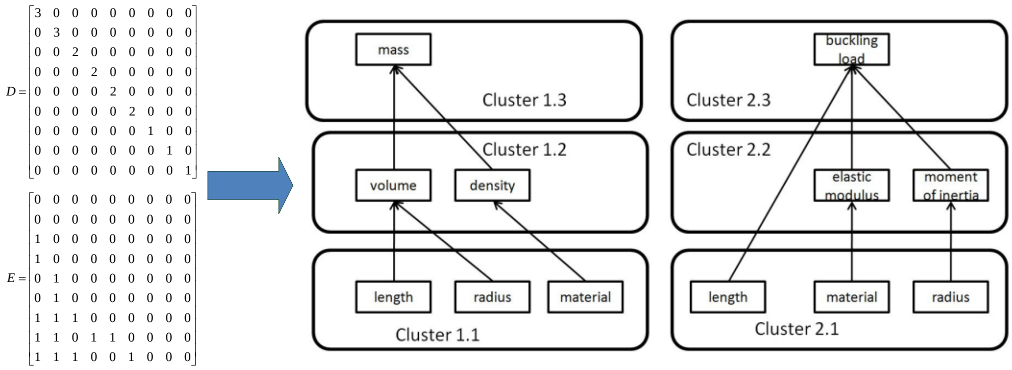

Figure 5. Clustered graph

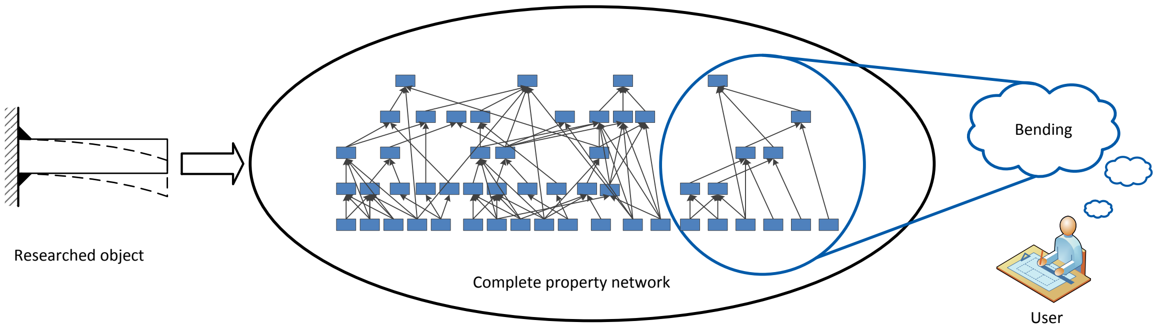

Figure 6. Transformation of properties into requirements

Aside from the requirements activated automatically by their experience, the user can always choose to manually look at more or fewer levels of the sub-tree. This user behaviour could be monitored to identify discrepancies between customer behaviour and the initially assigned experience value to alter this value accordingly. The criterion for the area of responsibility is the department the user works in.
The definition of areas of responsibility ensures that requirements are only activated for users with an understanding of the area the requirement belongs to. This prevents the definition of incorrect requirement due to a lack of expertise. Figure 7 shows a schematic representation of the envisaged procedure, which is explained in detail in the next sections.

3.2.1 Attribution of areas of responsibility

Aside from a few requirements defined by the customer, most requirements inside the solution space should be defined by the development team, as stated in Section 2. These requirements are not necessarily defined only by the development section of the company. Requirements can also arise from other departments of the company. To only activate requirements in the field of competence of the users of the department, every requirement needs to be assigned to one or more areas of responsibility. This is done using a method developed by Hanusch (Hanusch 2011) for the examination of responsibilities for sustainability in companies that was expanded to enable the examination of product properties in general. For each phase of the product life cycle the individual processes are assigned to the departments that influence the process: this can be depicted using swim-lanes (Hanusch 2011). The swim-lanes created this way have a universal validity and can be used to assign areas of responsibility to processes during the product life cycle. Therefore, it is possible to assign areas of responsibility to a property by using the swim-lanes if the processes that influence the property during the product life cycle are known.

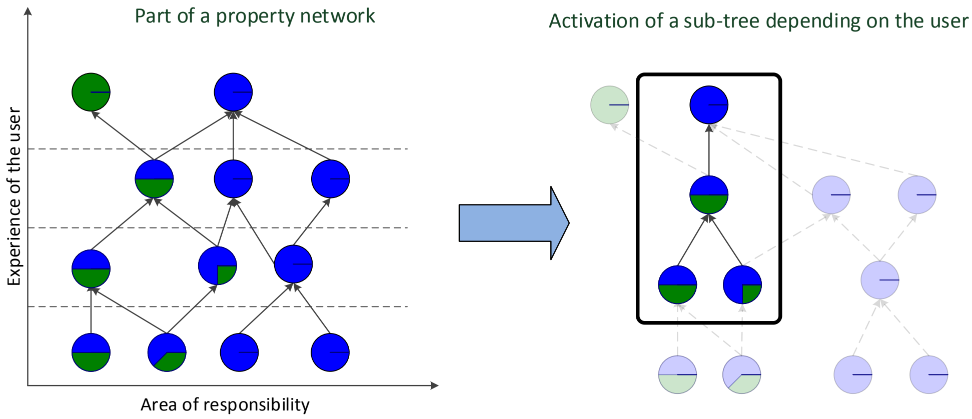

Figure 7. Activation of requirements depending on the respective user

If more than one department is responsible for a property the responsibility can be weighted based on how many of the processes affecting the property are assigned to the respective department. This weight allows a percentage of responsibility for a requirement to be given to the various departments. This has two advantages over a simple "is responsible/is not responsible" distinction. A percentage value for responsibility can be depicted in a pie chart. This gives the user a visual indicator of whether or not the selected requirement is of particular importance to their department and which department should be contacted if questions or problems regarding this requirement arise. In addition, assigning responsibilities to every property represents a major effort. Due to this fact, responsibilities are only assigned to a few properties and the responsibility of other properties is assigned by using the method of inheritance described below. In the case of a yes/no decision about responsibility, a method of inheritance would result in requirements decided by many departments who are all equally responsible. This would make the decision-making process very difficult. By using a responsibility value instead it is possible to define a threshold value above which requirements become activated for a department. To reduce the effort needed to assign responsibilities to all properties within the network, two different methods of inheritance were developed. The idea of these methods is to only assign responsibilities either to the sources or the sinks of the initial property network and determine the responsibilities for the other properties by passing on the responsibilities in the direction of the edges respectively against that direction. If a property inherits responsibilities from different sources the average value is set as its responsibility value for the individual departments. Whether the sources or sinks of the network should serve as a starting point depends on the aim of the attribution. If started at the sinks of the network, the attribution will usually result in a shift of responsibility towards the departments of research and development as well as the marketing. This is due to the fact that the sinks are often requirements defined by the market or target properties that need to be optimized by the development department. Since the development department usually has the main responsibility for the requirement acquisition process, the generated responsibilities mentioned above can be used for the requirement acquisition. If the attribution starts at the sources of the network, the resulting responsibilities mainly depend on independent product properties. This causes a shift in the responsibilities towards the production department since they are directly involved in the actualization of the properties in the final product. This attribution of responsibilities is based on properties that are not necessarily important to the success of the product. However, it can be used to identify restrictions resulting from the production processes of the company of which the development engineer might not be aware. Examples of this could be available but would not work for capacity machinery or the dimensions of doors or other bottlenecks the product has to fit through during the production.

4 EXAMPLE OF USE

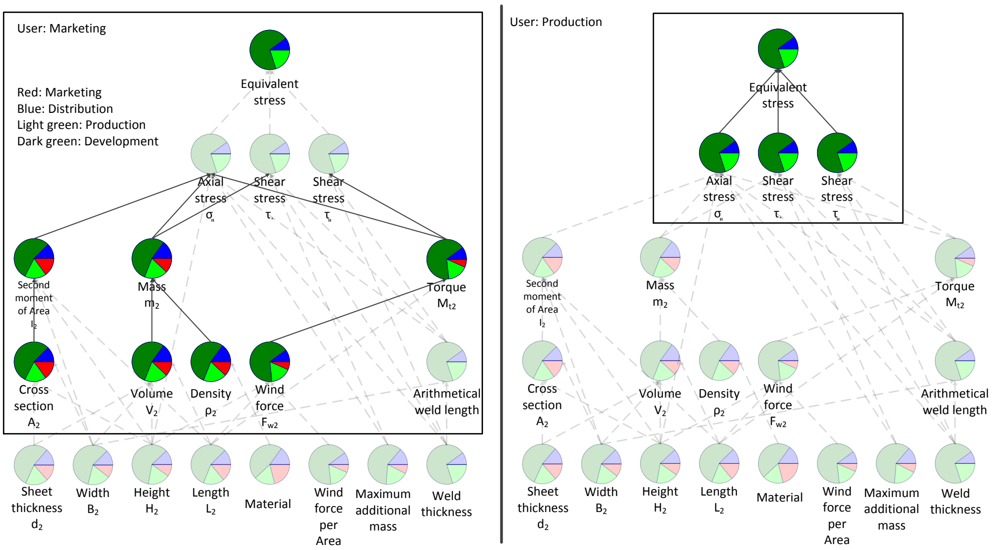

Figure 8. Example of requirement activation by different users

The developed method was used to examine a property network belonging to a street light consisting of two parts welded together. Figure 8 depicts the sub-tree of the equivalent stress inside the welding seam and the activation of additional requirements if a requirement for the equivalent stress is defined. On the left side of the figure the activation for an experienced user from the marketing department is shown. The right side of the figure shows the activation for an inexperienced user belonging to the production department. While the experienced user is confronted with more requirements that are also situated in deeper levels of the sub-tree, the number of requirements activated for the inexperienced user is considerably smaller. This ensures that an inexperienced user is not overwhelmed by the amount of information given simultaneously. Additionally, the limitation to activate only requirements in the upper levels of the sub-tree prevents users without the experience needed from defining incorrect requirements in lower levels whose influence on other areas of the property network the user might not be aware of. In addition to the activation based on the experience of the user, a preselection of requirements is made based on the area of responsibility of the user. This prevents requirements from being activated that the user from the marketing department might lack the technical expertise to comprehend. This can prevent incorrect requirements that could occur due to misinterpretation of a requirement by the user. By using the user-specific selection of requirements based on experience and area of responsibility of the user it is possible to improve the process of requirement acquisition by considerably reducing the amount of requirements shown to the user at the same time. It allows the user to select from a small amount of activated requirements while being able to comprehend their connection between one another, which is a lot easier than identifying the needed requirement in the complex depiction of the initial property network. In addition, it is possible to use a system to test the requirements defined within the sub-tree for consistency with the rest of the property network and report occurring conflicts to the user.

5 CONCLUSION AND FURTHER WORK

The method to transform property networks into requirement clusters uses a systematic approach to create an objective and complete knowledge database. The usage of this knowledge database for requirement acquisition, as described in this paper, helps the user create a complete and correct requirement list. The use of mathematical methods to divide the property network into clusters based on the structure of the initial network presents a systematic and automatable way of creating objective requirement clusters. The resulting objective clusters can be used to improve the internal standardization of CRC 666. The method is not bound to a specific topic within mechanical engineering and can be used for any given property network. In addition, it was shown that the resulting clusters can be used to aid in the acquisition of requirements. With the described method for requirement acquisition it is possible to reduce the necessary iterative steps due to the possibility of detecting conflicting requirements using the connections within property networks. The activation of additional requirements benefits the acquisition of implicit requirements that might not otherwise be noticed. Additionally, the activation and examination of requirements belonging to a sub-tree allows significant simplification of the work with otherwise complex property networks. The connection between requirements already known from the initial property network ensures that already defined requirements and their influence on the product are not forgotten during the later stages of development. Additionally, due to the envisaged computer-based implementation of this method, it would be possible to create forms for communication between users that ensure consistent and precise formulations to prevent misconception. However, there are still some challenges that need to be dealt with and conditions that have to be fulfilled to actually use the developed method. Since the requirements that can be acquired by the developed method are limited to those that already exist within the initial property network a complete and correct initial network is vital for a successful requirement acquisition. The objective requirement clusters created by using property networks only are one part of the whole cluster catalogue mentioned in Section 2. Additionally, different methods for the activation of requirements, for example, by surrounding factors or the product itself, should be examined. Finally, it is necessary to find a way of creating solution-neutral requirement clusters that can be used by the customer to define the problem space described in Section 2.

REFERENCES

Gramlich, S. (2013) *Vom fertigungsgerechten Konstruieren zum produktionsintegrierenden Entwickeln - Durchgängige Modelle und Methoden im Produktlebenszyklus*, Düsseldorf, VDI-Verlag

Hanusch, D. (2011) *Die Entwicklung nachhaltiger Produkte - Theoretischer Anspruch und Möglichkeiten zur praktischen Umsetzung in der sozialen Dimension*, Düsseldorf, VDI-Verlag

Heidemann, B. (2001) *Trennende Verknüpfung: ein Prozessmodell als Quelle für Produktideen - Produktentwicklung Aufgabenklärung Konzept Forschung Hilfsmittel Methoden Modelle*, Düsseldorf, VDI-Verlag

Normenausschuss Qualitätsmanagement, Statistik und Zertifizierungsgrundlagen im DIN (2005) *DIN EN ISO 9000:2005-12 Qualitätsmanagementsysteme -Grundlagen und Begriffe*, Berlin, Beuth Verlag

Pahl, G., Beitz, W., Feldhusen J, and Grote, K.-H. (2007) *Konstruktionslehre - Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung, 7.Aufl.*, Berlin, Heidelberg, New York, Springer-Verlag

Röder, B., Birkhofer, H., und Bohn, A. (2011) 'Clustering customer dreams - An approach for more efficient requirement acquisition', *International conference on engineering design ICED11*, Copenhagen, 2011, 10

Sauer, T. (2006) *Ein Konzept zur Nutzung von Lösungsobjekten für die Produktentwicklung in Lern- und Anwendungssystemen*, Düsseldorf, VDI-Verlag

Schott, H. (1997) *Informationsressourcen und Informationsmanagement für die Entwicklung umweltgerechter Produkte*, Düsseldorf, VDI-Verlag

Turau, V. (2009) *Algorithmische Graphentheorie, 3. Aufl.*, München, Oldenbourg Wissenschaftsverlag

Wäldele, M. (2012) *Erarbeitung einer Theorie der Eigenschaften technischer Produkte - Ein Beitrag für die konventionelle und allgorithmenbasierte Produktentwicklung*, Düsseldorf, VDI-Verlag

Wäldele, M., Chahadi, Y., Birkhofer, H. (2008) *Von der Kundenanfrage zur standartisierten Produkteigenschaft - Ein algorithmenbasierter Ansatz, Tagungsband 2*, Zwischenkolloquium Sonderforschungsbereich 666