

SIMULTANEOUS DEFINITION OF KEY CHARACTERISTICS IN ORDER TO FACILITATE ROBUST DESIGN IN EARLY PRODUCT DEVELOPMENT STAGES

**Goetz, Stefan;
Horber, Dennis;
Schleich, Benjamin;
Wartzack, Sandro**

Friedrich-Alexander-Universität Erlangen-Nürnberg

ABSTRACT

The success of complex product development projects strongly depends on the clear definition of target factors that allow a reliable statement about the fulfilment of the product requirements. In the context of tolerancing and robust design, Key Characteristics (KCs) have been established for this purpose and form the basis for all downstream activities. In order to integrate the activities related to the KC definition into product development as early as possible, the often vaguely formulated requirements must be translated into quantifiable KCs. However, this is primarily a manual process, so the results strongly depend on the experience of the design engineer.

In order to overcome this problem, a novel computer-aided approach is presented, which automatically derives associated functions and KCs already during the definition of product requirements. The approach uses natural language processing and formalized design knowledge to extract and provide implicit information from the requirements. This leads to a clear definition of the requirements and KCs and thus creates a founded basis for robustness evaluation at the beginning of the concept design stage. The approach is exemplarily applied to a window lifter.

Keywords: Robust design, Conceptual design, Requirements, Early design phases

Contact:

Goetz, Stefan
Friedrich-Alexander-Universität Erlangen-Nürnberg
Engineering Design
Germany
goetz@mfk.fau.de

Cite this article: Goetz, S., Horber, D., Schleich, B., Wartzack, S. (2021) 'Simultaneous Definition of Key Characteristics in Order to Facilitate Robust Design in Early Product Development Stages', in *Proceedings of the International Conference on Engineering Design (ICED21)*, Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.530

1 INTRODUCTION

In today's dynamic and competitive markets, an increasing demand for quality and complexity requires a comprehensive but flexible definition of product requirements as starting point for product development. The diverse characteristics of the large number of requirements inevitably lead to unclear or ambiguous requirement definitions. Nevertheless, the technical realization calls for a suitable interpretation and their translation into appropriate target factors to fulfil customers' demands. In the context of robust design and tolerancing, these factors are usually called Key Characteristics (KCs) forming the basis for subsequent activities (Thornton, 2004). Even though their definition plays a crucial role in product development, it is usually a manual process, whose results strongly depend on the design engineer's expertise. Especially in the context of a dynamic product development process, this step is time consuming and thus a hurdle for an early robustness evaluation of principal solutions. Therefore, this contribution presents a knowledge-based approach that utilizes natural language processing (NLP) for the simultaneous derivation of functions and KCs during requirement definition. The paper is structured as follows. Section 2 discusses related work considering requirements engineering and design evaluation in the context of early robust design. Based on the resulting research question in section 3, section 4 describes the proposed approach, which is exemplarily applied and discussed in section 5. Finally, the paper closes with a conclusion and an outlook.

2 RELATED WORK

Since product development processes usually start with the definition of requirements, it has a decisive influence on design decisions and their traceability. Thus, the requirements engineering including the identification and unification of stakeholder requirements and their appropriate documentation and management emerged. (Pohl, 2010) Due to the various sources of requirements and the progressive concretization along the product development process, there are several types (e.g. process or product requirement) and levels of detail (Dick *et al.*, 2017). For example, a strict distinction between wishes and demands is drawn (Rupp, 2014). Apart from this characterization, the quality of the requirement definition particularly affects the product success (Kamata and Tamai, 2007). Thus, a proper documentation of this various information is indispensable (Pohl, 2010). Besides a formal structured documentation, for example via UML or SysML, textual requirements in sentences, e.g. in requirement lists, are preferred in the design engineering context (INCOSE, 2017). In order to ensure a high quality, for example by avoiding redundant, unclear and ambiguous requirements (Dick *et al.*, 2017), there are extensive writing guides leading to a proper wording of the requirement definition (INCOSE, 2017). However, ambiguities in requirements definition cannot be completely prevented by these guidelines, especially in large development teams. Thus, particularly in the context of software development, NLP is increasingly used to systematically analyze, classify and improve the requirement definition (Nazir *et al.*, 2017). This includes the elimination of ambiguities from text (Kiyavitskaya *et al.*, 2008) and the transformation into design artefacts (Yue *et al.*, 2011). Despite extensive preliminary work in the area of using NLP for requirements engineering, the adaptation to individual contexts or applications is still part of ongoing research (Dalpiaz *et al.*, 2018; Zhao *et al.*, 2020).

Once the requirements are properly defined, they are translated into specific design parameters in the ongoing development process according to the Axiomatic Design (Suh, 1990). Thereby the House of Quality (HoQ) from the Quality Function Deployment (QFD) facilitates the structured, matrix-based mapping of the relations between requirements and design parameters (Sullivan, 1986). This breakdown constitutes a significant concretization of the product and is therefore often time-consuming. Nevertheless, this step is essential at the beginning of the product development process, especially with regard to a robust product design (Göhler *et al.*, 2016). In this context these parameters, which map the requirements in a quantifiable way, are usually called Key Characteristics (KCs) (Thornton, 2004). They are the basis for tolerancing and robust design activities and improve the traceability of design decisions throughout the product development process (Zheng *et al.*, 2008). Moreover, they enable an early quantitative design evaluation and thus contribute to a largely objective concept selection in the next step of product development (Okudan and Tauhid, 2008). Accordingly, a quantitative approach for the evaluation of principal solutions under consideration of different robustness principles was presented. The basis for this approach is the function structure of the product as well as the derived KCs, which are included in the evaluation with a weighting based on the Variation Risk Priority Number (VRPN) from a modified Variation Mode and Effect Analysis

(VMEA). This leads to one individual robustness index for each principal solution enabling the consideration of the aspect of robustness in the multi-criteria design evaluation at the beginning of the concept design. (Goetz *et al.*, 2019)

3 RESEARCH QUESTION

Summarizing the related work, the proper definition of requirements is crucial for successful product development and reliable design evaluations. Besides the formalization of the requirements, this calls for a translation into design parameters enabling the concretization from implicit to explicit product information in the ongoing development process. However, there is a lack of specific approaches supporting the design engineer in this step. Thus, the actual linking of requirements and quantitative parameters is often weak, which is a major burden especially in the concept evaluation (Okudan and Tauhid, 2008). Accordingly, the results of the robustness evaluation of principal solutions significantly depend on the requirement-based definition of KCs and thus on the experience as well as the effort of the respective design engineer. Consequently, motivated by the demand for a more reliable and accelerated derivation of KCs the following research question arises: How can ambiguities in the definition of criteria for robustness evaluation in early design stages be consistently reduced?

4 SIMULTANEOUS DEFINITION OF KEY CHARACTERISTICS

In the common procedure, requirements are sequentially broken down to specific KCs, see Figure 1. However, this one-way procedure is subjective and leads to time-consuming iterations in case of requirement changes. In contrast, the proposed approach enables a simultaneous definition of requirements, functions and KCs fostering a better linkage, see Figure 1. Moreover, the utilization of NLP and formalized design knowledge enables an automatic classification of requirements leading to a clear understanding of the intention of the requirement and a computer-aided derivation of functions, KCs and their attributes. The classification uses the classes necessity (wish/demand), aspect (qualitative/quantitative) and condition (hurdle/optimization) defined in preliminary work (Horber *et al.*, 2019). Since this classification focuses on the automated derivation of evaluation criteria (Horber *et al.*, 2020), it is reasonable for the intended robustness evaluation.

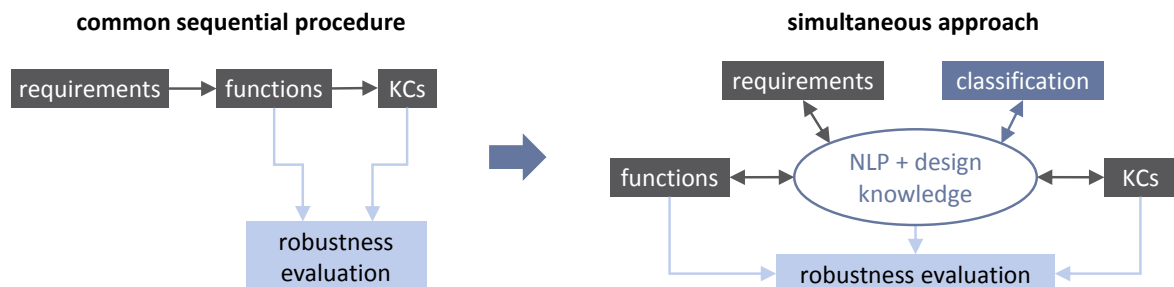


Figure 1. Transition from sequential to simultaneous KC definition

After a brief introduction of the corresponding workflow shown in Figure 2, the principles of the approach are explained in detail in the subsequent paragraphs. The process starts with an initial requirement definition, which is automatically classified revealing what is commonly meant by this specific requirement. This enables a stringent adaption of the wording by the design engineer so that the actual intention becomes clear. Simultaneously, the automatic derivation of function and KC proposals with associated attributes such as target values and directions of improvement takes place. The adoption of the function proposals and the corresponding weighting with respect to the relevance for the associated requirement leads to a proper function definition influencing the derived KCs. Their adapting and weighting according to the VMEA presented by Goetz *et al.* (2019) completes the simultaneous definition of KCs. Finally, the results from the proposed approach are unified in one common modified HoQ including requirements, functions, KCs and the corresponding weightings and relations. This documentation of relevant information for a specific product allows a comprehensive consistency check that helps to avoid redundant or ambiguous definitions. Moreover, it forms the basis for the robustness evaluation of principal solutions according to Goetz *et al.* (2019).

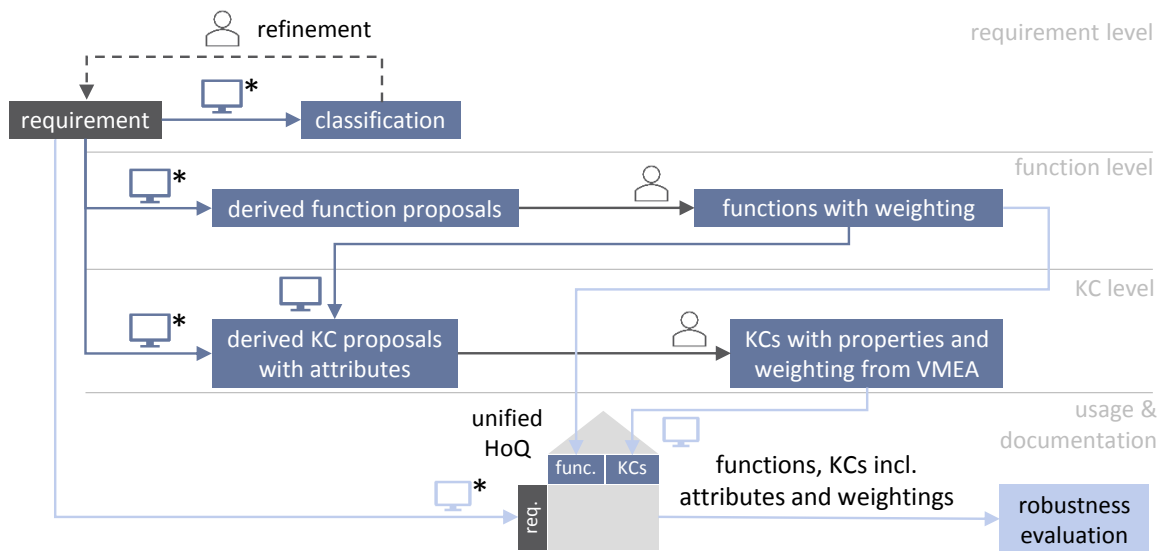


Figure 2. Workflow of simultaneous function and KC definition and its documentation. Pictograms indicate automated steps and design engineer interaction. Simultaneous steps are marked with an asterisk.

The automated approach is based on fundamental NLP features shown in Figure 3. In combination with rule-based matching (Nadkarni *et al.*, 2011) this allows the integration of formalized semantic design knowledge that is especially necessary in case of less specific requirements. In order to cover a wide range of requirements with different level of detail, the proposed approach combines various principles. These different aspects and their realization are explained below by means of **three types of requirements** with decreasing information content.

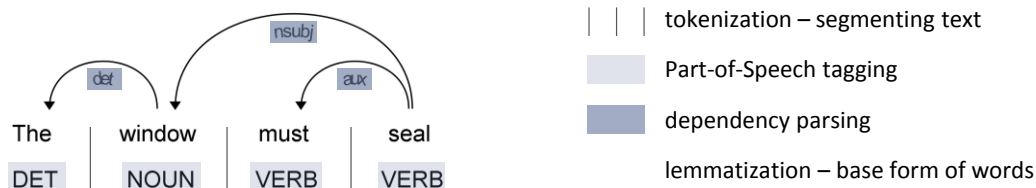


Figure 3. Explanation of basic NLP features (generated with spaCy)

Requirement with direct description of KC

In order to identify this requirement type, it is first compared with the appropriate defined KC list including the terms *gap*, *flush*, *distance*, *clearance*, *angle*, *tilt* and *tilting* commonly used for the explicit description of KCs. In combination with dependency parsing, this allows the extraction of the **KC** including the corresponding components, see Figure 4.

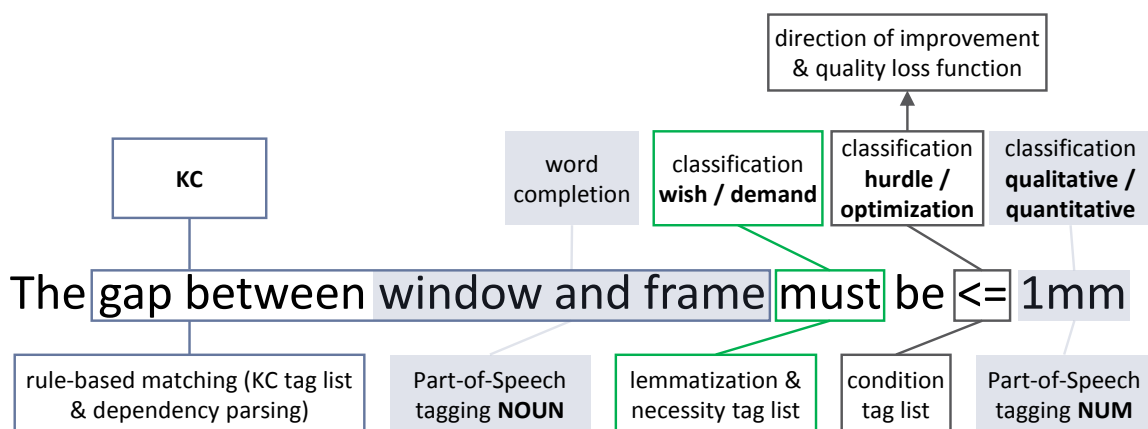


Figure 4. Extracted information (upper section) and corresponding techniques (lower section) for exemplary direct requirement definition

Furthermore, the identification of nouns within the requirement specification enables the realization of a word completion approach. This accelerates the subsequent definition of further requirements and ensures consistent word usage.

Based on the fundamental idea described in (Horber *et al.*, 2020) the proposed extension of the classification of requirements allows a detailed derivation of the requirements intention. First, the existence of a numeric modifier (NUM) in a requirement indicates whether it is **qualitative or quantitative**. The distinction between **wish and demand** is realized by matching the lemma of the verb assigned to the KC with a tag list. Thereby terms like *should, could, can or might* indicate wishes while the words *must, need to, have to, shall or may* describe a demand. Moreover, the tag list in Figure 5 enables a classification in **hurdle and optimization**. This helps to distinguish whether a requirement respectively evaluation criteria is already completely fulfilled by exceeding or falling below a threshold value (hurdle) or whether further improvements are useful (optimization). Even though this classification is beneficial in the design evaluation process (Horber *et al.*, 2020), detailed information about the desired behavior of a product under variation is missing. Therefore, the table in Figure 5 additionally provides quality loss functions (Phadke, 1989) that are commonly used in the context of robust design. They enable a proper evaluation of design alternatives during the deviation and robust design analysis in the subsequent product development steps. Finally, the derivation of the direction of improvement allows its proper documentation in the HoQ (Sullivan, 1986).

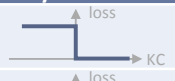
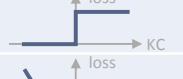
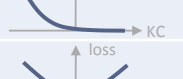
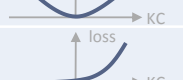
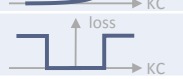
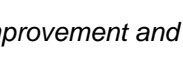
tags	condition	direction of improvement	quality loss function
larger, greater, more, at least, minimum, minimal	hurdle	maximize	
shorter, smaller, fewer, less, at most, maximum, maximal	hurdle	minimize	
>, >=, =>	optimization	target	
=, as possible, equal			
<, <=, =<			
combination of "maximize" and "minimize"			

Figure 5. Condition tag list with corresponding condition, direction of improvement and quality loss function

Requirement with indirect description of KC

Besides the direct definition of KCs with corresponding target values, requirements often provide an indirect description of KCs. Therefore, a correct interpretation and transfer into an explicit definition is essential. Unlike the mostly manual and individual procedures, the proposed approach utilizes semantic and formalized knowledge for terms that are typically used. Figure 6 shows an extract of these tags with corresponding information.

tags	KC term	target value
parallel	angle	0°
perpendicular, rectangular		90°
coincident, concentric	displacement	0mm

Figure 6. Exemplary tag list indicating KCs and target values

Thus, for example, the requirement *"The horizontal gap between window and frame must be as parallel as possible"* leads to a proposed KC *angle gap window frame* with target value 0°. Furthermore, in combination with the principles shown in the previous paragraph, the requirement is automatically classified and further attributes, such as the direction of improvement, are derived.

Requirement defining a function

Finally, requirements often refer to specific functions of a product. Since a direct link to KCs is not apparent, these requirements are frequently disregarded during the prevailing manual determination of KCs. However, to ensure these functions, specific KCs must be observed. Thus, a knowledge-based

approach, which automatically and uniformly derives associated KCs, is beneficial to ensure that implicitly necessary KCs are considered in further development. As the excerpt in Figure 7 shows, a largely universally valid knowledge base is generated with the aid of engineering design logic and formalized empirical knowledge. For example, the function *seal* is usually associated with a gap, which should be kept to a minimum if possible.

verb	KC term	direction of improvement
guide	clearance	minimize
seal	gap	minimize
not jam	clearance	maximize

Figure 7. Exemplary tag list for functions with correspond. KC and direction of improvement

For instance, the requirement "The window must seal" stringently leads to the function *seal window*, the KC *gap*, the direction of improvement *minimize* and the corresponding quality loss function. Thus, the proposed approach classifies requirements with varying degrees of detail and automatically derives associated functions and KCs with the respective attributes. Finally, KCs are broken down into individual KCs according to the six degree of freedom (DOF), which allows a separate consideration in the subsequent steps. For example, a distinction between translational and rotational clearance is useful. The documentation of these abstracted KCs in the unified HoQ enables an easy check for redundant KCs and thus helps to avoid ambiguous definitions. Moreover, motivated by the contradiction analysis for functional requirements (Göhler and Howard, 2015), conflicting KCs and thus requirements are simply avoided or combined by considering the direction of improvement and the target value for multiple defined KCs in the extended HoQ. For better traceability and easy identification of weak points, the relations are additionally represented in a graph, see the exemplary excerpt in Figure 8.

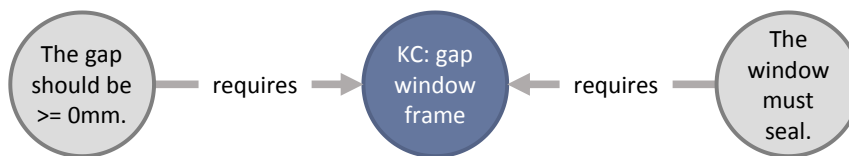


Figure 8. Exemplary graph with multiple definition of a KC

5 APPLICATION

The proposed approach is exemplarily applied to a simplified electric car window regulator to demonstrate the practical workflow. This academic case study is industry-oriented and offers various concept alternatives such as the cross-arm or dual rail cable mechanism enabling a useful robustness evaluation (Goetz *et al.*, 2020). Moreover, the adequate simplicity of the system allows for good comprehensibility. The requirements listed below are exemplarily defined and do not claim to be comprehensive. They are partly redundant or ambiguous and formulated in different detail, see for example requirement 3 and 5. This initial definition is the starting point for the application of the approach.

1. The window shall be guided properly
2. The window should close quickly
3. The window must seal
4. The window must be firmly attached to the mechanism
5. The gap between window and frame must be $\leq 1\text{mm}$
6. The window must not jam
7. The window must not transfer excessive tension
8. The gap between window and frame must be $\geq 0\text{mm}$
9. The horizontal gap between window and frame must be as parallel as possible

5.1 Implementation

The conceptual implementation is based on spaCy, which is an open-source library enabling NLP. Figure 9 shows the prototype of the associated graphical user interface (GUI). In compliance with the

structure of the workflow presented in Figure 2, the GUI is organized in three levels: requirement, function and KC. Consequently, the information belonging to an individual requirement is summarized in one single view.

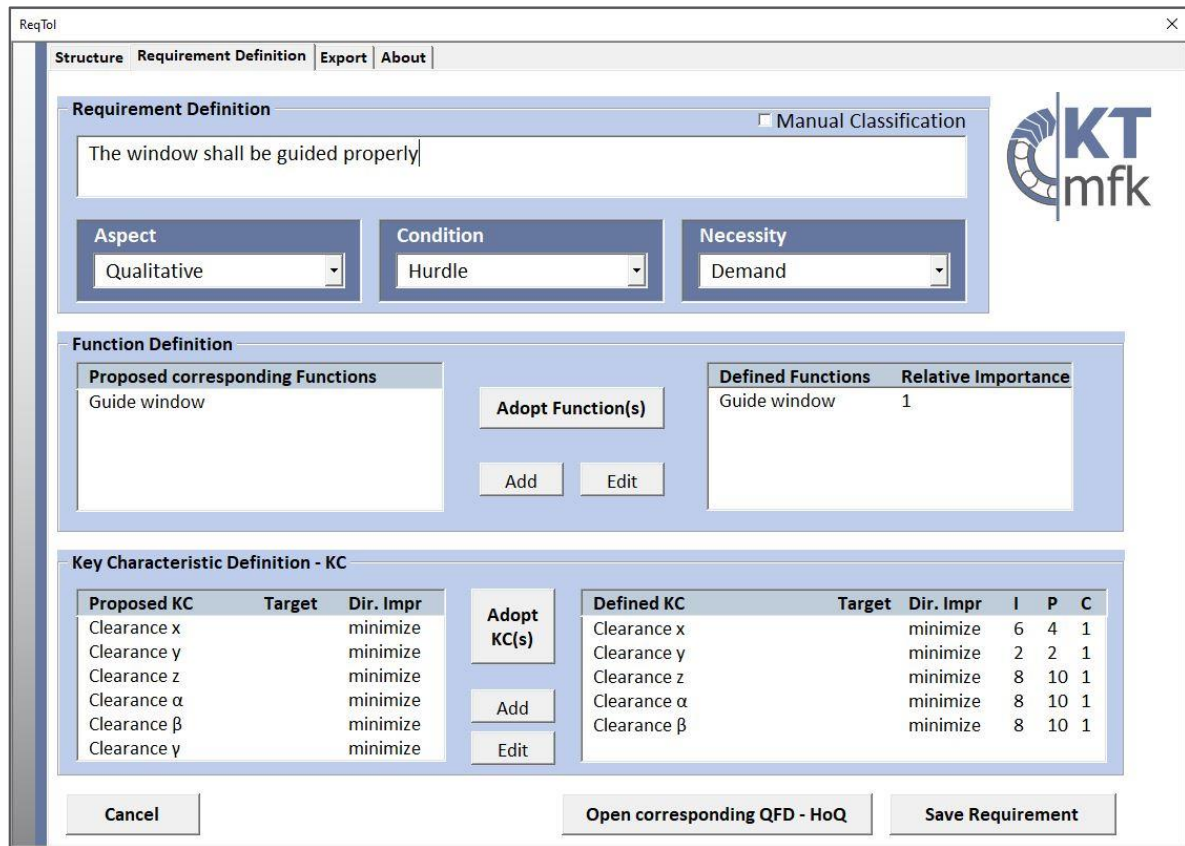


Figure 9. Prototype of GUI combining requirement, function and KC level

In the first step, the respective requirement is initially defined by the user, processed in the backend via NLP and automatically classified. If necessary, the design engineer can further specify the requirement so that the actual intention becomes clear and matches his understanding. Modifications of the requirements directly affect the resulting proposals for functions and KCs.

In the center of the GUI in Figure 9, the associated functions are simultaneously defined. The design engineer has the option to adopt and modify the automatically generated proposals. In order to avoid redundancy in the definition of further requirements, a comparison with the entries of the HoQ identifies if the function has already been determined. Finally, the design engineer is asked to weight the respective functions with values between 0 and 1 reflecting its relative importance for the superior requirement. This corresponds to the entries of the relation matrix of the HoQ and forms the basis for the structured robustness evaluation (Goetz *et al.*, 2019).

In the lower section, KCs are defined based on the proposed KCs including the corresponding directions of improvement and, if applicable, the target values. According to the information from the tag list in Figure 7, clearance for all six DOF is proposed, see Figure 9. The proposal also respects the check for already specified redundant or conflicting definitions. Since the design engineer considered the clearance γ as irrelevant, five KCs associated with the requirement are determined. Finally, these KCs are weighted using the modified VMEA (Goetz *et al.*, 2019). This includes the assessment of the importance (I) of the KC for the superior function or requirement, its probability (P) of variation and, if applicable, a correction(C) factor.

This process is repeated for each requirement. The information, which is defined in the GUI, is stored in an Excel-based HoQ, see Figure 10. This comprises all requirements, the associated functions as well as KCs and forms the basis for the proposed consistency check. Thus, for example, the two requirements "The window shall be guided properly" and "The window must not jam" were merged because they have a similar intention, which results in the derivation of the same functions and KCs. However, since the direction of improvement of the KCs for *guide* and *not jam* is opposing (see Figure 7), their unification

led to a new quality loss function, see Figure 5. This proposed consistency check enabled the elimination respectively combination of two requirements and seven KCs for the case study presented here. Moreover, the HoQ shows the interrelations. The roof of the HoQ represents the relations among functions and KCs. The relation matrix in the center shows the relative importance of the functions and the resulting product of I, P and C from the GUI. The multiplication of these values in the relation matrix according to the relations defined in the roof results in the VRPN indicating the importance of the KC for the entire product. This allows their easy prioritization for the subsequent product development steps.

Direction of Improvement		▼	▼	▼	▼	▼
Functions, Subfunctions, KCs	Guide window	Clearance x	Clearance y	Clearance z	Clearance alpha	Clearance beta
Requirements						
The window shall be guided properly	1	24	4	80	80	80
Target						
VRPN (Technical Importance)		24	4	80	80	80
Relative Weight		9%	1%	30%	30%	30%

Figure 10. Extract from unified HoQ for requirement 1 with information defined in the GUI

Thus, the HoQ forms the basis for the Excel-based robustness evaluation matrix shown in Figure 11. The information adopted from the previous steps are highlighted in blue. Consequently, the task of the design engineer is reduced to the concept evaluation regarding different robustness criteria in the center of the evaluation matrix. In contrast to the procedure described in (Goetz *et al.*, 2019), the design engineer is supported by additionally derived information such as quality loss functions and, if applicable, target values. The detailed process of robustness evaluation is extensively described in (Goetz *et al.*, 2019). So, the exemplary evaluation in Figure 11 does not claim to be comprehensive and is primarily used for demonstrating the benefit of the proposed approach here. Based on the resulting indices, the principal solution dual rail cable window regulator is more robust than the cross-arm window regulator (Goetz *et al.*, 2020).

Input - HoQ		Key Charakteristic - KC						Σ-values	Σ-VRPN	Ø-cross arm	Ø-dual rail	
Calculate	weighting	FM	clear z	clear α	clear β	gap	angle gap					
criteria		qual. loss										
		VRPN	80	80	80	80	60					
degree of constraint	3	cross arm	1	2	2	2	2	2040	380	5.4		
		dual rail	1	1	1	2	1	1380			3.6	
number of parameters	2	cross arm	1	2	2	2	1	1240			3.3	
		dual rail	1	2	1	1	1	920				2.4
product - functions (guide & move window)												
coupling	2	cross arm	3					6	-	6		
		dual rail	1					2			2	
										robustness index (cross arm) 14.7		
										robustness index (dual rail) 8		

Figure 11. Robustness evaluation matrix for two window lifter concepts

5.2 Discussion

As the matrix in Figure 11 shows, the robustness evaluation is simplified by the information provided by the proposed approach. Although the evaluation criteria are automatically provided, the results still depend on the designers' experience. However, the remaining ambiguity in the definition of these criteria is reduced by the additional available information, such as the quality loss functions. Thus, the research question is answered by using NLP and tag lists with semantically linked design knowledge deriving implicit information from the requirements. Therefore, the requirement definition must follow rules. This issue is countered by the semantic analysis of the requirements, which extracts their intention e.g. by classification. Thus, the intended interaction with the design engineer stringently improves the requirement definition and leads to a uniform understanding, without automatic changes of the input. The proposed approach uses rule-based NLP, in which the information associated with a requirement is explicitly mapped. As typical for knowledge-based approaches, there is a contradiction between the degree of detail and the universal applicability of the formalized design knowledge in the tag lists. Its extent depends on the scope and unambiguity of the requirement definition. Especially within a company with recurring requirements, it allows comprehensive automation. However, confirmation by the design engineer is essential at this early stage, where the design space is extensive. The simultaneous NLP-supported definition of requirements, functions and KCs reveals intensions and relations so that together with the designer's interaction a consistent definition can be ensured. This clarity, as well as the linked documentation, is particularly beneficial for complex products with numerous interacting requirements or dynamic changes, as their impact is immediately clear. However, the degree of automation, especially in the derivation of implicit information, depends on the formalized knowledge and the level of detail of the requirement definition.

6 CONCLUSION AND OUTLOOK

The contribution stresses the lack of specific computer-aided approaches supporting the design engineer in the translation of differently formulated requirements into quantifiable KCs, which are the basis for tolerancing and robust design decisions. Since these KCs are the only quantitative criterion available for robustness evaluation at the beginning of the conceptual design, their clear definition is essential. Thus, the novel approach fosters the simultaneous definition of requirements, functions and KCs supported by the semantic analysis and derivation of information from the requirements. The implicitly available information is translated into concrete specifications with the help of NLP, rules and formalized design knowledge. Besides, the classification of the requirements contributes to a clear understanding of their meaning. Thus, the unambiguity and quality of the requirement is consistently improved. Furthermore, the comprehensive semantic analysis does not only derive functions and KCs but also additional information such as associated quality loss functions. This extended information supports the design engineer in the subsequent robustness evaluation.

Moreover, the approach provides an intensive linkage between requirements, functions and KCs, which enhances traceability of the effect of requirement changes. These relations are documented in an extended HoQ unifying the relevant information. The common document enables a consistency check already during the definition of requirements, which prevents redundant or contradicting requirements and KCs. Finally, the HoQ also covers the weightings of the functions and KCs regarding their importance for the entire product so that all input variables relevant for the robustness evaluation of concepts are available. Apart from the improvement of the applicability of the early robustness evaluation, the proposed approach contributes to a well-founded requirement definition.

To further exploit the potential of the proposed approach, a future integration into the multi-criteria evaluation process of concepts would be useful. In this context the derivation of further information, which is already implicitly defined in requirements, e.g. through machine learning is conceivable. Finally, the approach should be transferred to industrial application and adapted to the users' needs.

ACKNOWLEDGMENTS

The authors thank the German Research Foundation (DFG) for supporting the research projects 'Integrated computer-automated tolerance management in early design stages' and 'Multi-criteria evaluation in product development [MuBePro] – Computer-supported evaluation process of solution variants with regard to all requirements from the product life cycle' under the grant numbers SCHL 2233/1-3, WA 2913/17-3 and WA 2913/33-1.

REFERENCES

- Dalpia, F., Ferrari, A., Franch, X. and Palomares, C. (2018), "Natural Language Processing for Requirements Engineering: The Best Is Yet to Come", *IEEE Software*, Vol. 35 No. 5, pp. 115–119. <http://doi.org/10.1109/MS.2018.3571242>.
- Dick, J., Hull, E. and Jackson, K. (2017), *Requirements Engineering*, Springer International Publishing, Cham. <http://doi.org/10.1007/978-3-319-61073-3>.
- Goetz, S., Hartung, J., Schleich, B. and Wartzack, S. (2019), "Robustness Evaluation of Product Concepts based on Function Structures", *International Conference on Engineering Design*, 5.-8. August 2019, Cambridge University Press, pp. 3521–3530. <http://doi.org/10.1017/dsi.2019.359>.
- Goetz, S., Schleich, B. and Wartzack, S. (2020), "Integration of robust and tolerance design in early stages of the product development process", *Research in Engineering Design*, Vol. 31 No. 2, pp. 157–173. <http://doi.org/10.1007/s00163-019-00328-2>.
- Göhler, S.M. and Howard, T.J. (2015), "The Contradiction Index (CI): A New Metric Combining System Complexity and Robustness for Early Design Stages", *27th International Conference on Design Theory and Methodology*, 2.-5. August 2015, ASME. <http://doi.org/10.1115/DETC2015-47255>.
- Göhler, S.M., Husung, S. and Howard, T.J. (2016), "The Translation between Functional Requirements and Design Parameters for Robust Design", *Procedia CIRP*, Vol. 43, pp. 106–111. <http://doi.org/10.1016/j.procir.2016.02.028>
- Horber, D., Schleich, B. and Wartzack, S. (2019), "Ein Klassifizierungssystem zur Anforderungssystematisierung", *30th Symposium Design for X*, 18.-19. September 2019, pp. 227–238. <https://doi.org/10.35199/dfx2019.20>
- Horber, D., Schleich, B. and Wartzack, S. (2020), "CONCEPTUAL MODEL FOR (SEMI-) AUTOMATED DERIVATION OF EVALUATION CRITERIA IN REQUIREMENTS MODELLING", *DESIGN 2020*, 26.-29. October 2020, Cambridge University Press, pp. 937–946. <http://doi.org/10.1017/dsd.2020.52>.
- INCOSE - Requirements Working Group (2017), *Guide for Writing Requirements*. International Council on Systems Engineering, San Diego/USA
- Kamata, M.I. and Tamai, T. (2007), "How Does Requirements Quality Relate to Project Success or Failure?", *15th IEEE International Requirements Engineering Conference*, 15.-19. October 2007. pp. 69–78. <http://doi.org/10.1109/RE.2007.31>.
- Kiyavitskaya, N., Zeni, N., Mich, L. and Berry, D.M. (2008), "Requirements for tools for ambiguity identification and measurement in natural language requirements specifications", *Requirements Engineering*, Vol. 13, pp. 207–239. <http://doi.org/10.1007/s00766-008-0063-7>.
- Nadkarni, P.M., Ohno-Machado, L. and Chapman, W.W. (2011), "Natural language processing: an introduction", *Journal of the American Medical Informatics Association*, Vol. 18 No. 5, pp. 544–551. <http://doi.org/10.1136/amiajnl-2011-000464>.
- Nazir, F., Butt, W.H., Anwar, M.W. and Khan Khattak, M.A. (2017), "The Applications of Natural Language Processing (NLP) for Software Requirement Engineering - A Systematic Literature Review", *International Conference on Information Science and Applications*, 20.-30. March 2017, Springer, Singapore, pp. 485–493. http://doi.org/10.1007/978-981-10-4154-9_56.
- Okudan, G.E. and Tauhid, S. (2008), "Concept selection methods - a literature review from 1980 to 2008", *International Journal of Design Engineering*, Vol. 1 No. 3, pp. 243-277. <http://doi.org/10.1504/IJDE.2008.023764>.
- Phadke, M.S. (1989), *Quality engineering using robust design*, Prentice Hall, Englewood Cliffs, NJ.
- Pohl, K. (2010), *Requirements Engineering: Fundamentals, Principles, and Techniques*, Springer, Heidelberg.
- Rupp, C. (2014), *Requirements-Engineering und -Management*, Hanser, München.
- Suh, N.P. (1990), *The principles of design*, Oxford University Press, New York.
- Sullivan, L.P. (1986), "Quality function deployment", *Quality Progress*, Vol. 19 No. 6, pp. 39–50.
- Thornton, A.C. (2004), *Variation risk management: Focusing quality improvements in product development and production*, Wiley, Hoboken, NJ.
- Yue, T., Briand, L.C. and Labiche, Y. (2011), "A systematic review of transformation approaches between user requirements and analysis models", *Requirements Engineering*, Vol. 16 No. 2, pp. 75–99. <http://doi.org/10.1007/s00766-010-0111-y>.
- Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K. J., Ajagbe, M. A., Chioasca, E., Batista-Navarro, R. T. (2020) *Natural Language Processing (NLP) for Requirements Engineering: A Systematic Mapping Study*. [online] Available at: arXiv: 2004.01099v2 (02.03.21).
- Zheng, L.Y., McMahan, C.A., Li, L., Ding, L. and Jamshidi, J. (2008), "Key characteristics management in product lifecycle management: A survey of methodologies and practices", *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 222 No. 8, pp. 989–1008. <http://doi.org/10.1243/09544054JEM1045>.