A Common Home for Features and Requirements: Retrofitting the House of Quality with Feature Models

Emanuel Mätzler, Bernhard Wally and Alexandra Mazak
Business Informatics Group
Institute of Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstraße 9–11, 1040 Vienna, Austria
{maetzler, wally, mazak}@big.tuwien.ac.at

ABSTRACT

Quality function deployment (QFD) is a method for quality assurance developed for application in production processes. One prominent tool for implementing QFD is the *House of Quality* (HoQ), whose basic design principles have been left unchanged for the last decades. Modern concepts for handling product variability, most notably feature models, represent intuitive means for a refurbished roof construction of the HoQ, and thus more expressiveness in the definition of functional requirements.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*

General Terms

House of Quality, Feature Models, Requirements Engineering, Variability Modeling, Quality Function Deployment

1. INTRODUCTION

In the past few years markets have changed in terms of altered and multiplied customer requirements, partly driven by the ever growing availability of customer configurable and individualizable products, and to some extent driven by self-confident customers who demand certain functionality from producers. Customer requirements can vary, therefore ways have to be found to ensure that products satisfy customer requirements properly, but also engineering and management complexity should stay at a level that is still perceivable and manipulatable by human actors.

One method of describing customizable products can be what is known as *product line engineering* (PLE), with its inherent implications such as feature variability, configuration, etc.

VaMoS '15 , January 21–23 2015, Hildesheim, Germany
Copyright 2015 ACM 978-1-4503-3273-6/15/01 ...$15.00.
http://dx.doi.org/10.1145/2701319.2701334

A *feature model* (FM) is a well-known way to describe variability in product lines—according to Berger et al. FMs are by far the most prominent method to specify variability [5].

The *House of Quality* (HoQ) is a method to define relationships between customer desires (customer requirements) and technical characteristics. It is an approach to analyze how engineering decisions affect customer perceptions [11].

In this work we propose the combination of feature models and the House of Quality in order to leverage the syntax and expressive power of FMs for the definition of the "how" section of the HoQ. By using a formalized language for the definition of the interrelation of single features and feature groups, the analysis and comparison of variable products can be simplified: several product instances can be compared directly regarding the fulfillment of requirements. Furthermore the impact of selecting a single feature can be analyzed in detail.

The result of this this work is the introduction of the basic modeling approach in section 3. Furthermore we demonstrate two use cases: The first one describes how to evaluate a single product configuration (see section 3.1). In section 3.2 the comparison of two different product configurations is explained.

2. RELATED WORK

Feature modeling is a variability modeling approach introduced by Kang et al. for a structured *Feature-Oriented Domain Analysis* [13]. It has been improved and formalized over the last decades by e.g. converting FMs into propositional logic [3] and vice versa [9]. Binary feature models have been extended by concepts for cardinality-based feature models [18, 7, 4, 8]. Tools and concepts have been developed in order to re-use feature models in different configuration stages, such as configuration links [17]. Meta-information was introduced to feature models in order to allow for more complex configuration analysis and additional interpretation possibilities [6].

With *Clafer*, one of the most versatile and powerful frameworks for the definition and analysis of feature models and domain models has been presented. Clafer defines a powerful meta-model with first-class feature modeling support [2]. The analysis algorithm implemented in Clafer allows for multi-objective optimization in PLE [15, 1]—in that approach pareto-optimal products are derived with respect to quality objectives like cost, performance or security that are added as attributes to features [16].

In the *software reuse* research field, part of the focus was laid on domain analysis and engineering, which in turn relies on functional requirements engineering [10]. Marketing and product plans relying on reusable components organized in feature models are introduced in [14], relying on FORM (feature oriented reuse method) [12]. While feature modeling deals with functional requirements engineering in order to actually find out about features in a product line, the customer view has not been considered with an evaluatory aspect in mind, which is inherent to the House of Quality approach.

The House of Quality as introduced by Hauser and Clausing [11] is a method based on *quality function deployment* (QFD). A relationship matrix is used to relate customer needs (requirements) with artifacts to meet them (functional components). We are not aware of previous work dealing with the application of feature modeling on the House of Quality. Doing so enables a customer requirements viewpoint on feature modeling and thus a complementary source of information for e.g. customer based analysis of product variants.

3. VARIABILITY EVALUATION

By adapting the HoQ through combining it with feature modeling, we present a conceptual approach which brings together variability modeling and requirements engineering. The HoQ is a technique for correlating customer perceptions with engineering characteristics. The original HoQ, as shown in Figure 1, consists of a few basic elements: to the lower left are the "Customer Attributes", which are gathered by the customer and can be seen as the requirements. Since not every requirement is equally important for the customer, each requirement is weighted depending on the customer's perception (the chimney column: "Relative Importance"). These prioritized requirements are then the rows in the relationship matrix (to the lower right). The second dimension of that matrix describes the engineering characteristics defined by engineers and representing the columns of the matrix (cf. "Engineering Characteristics"). On top of this tower there is the roof of the house, which is also representing engineering characteristics: interrelations between technical characteristics are identified. In the original HoQ those relations are classified as "strong positive", "medium positive", "medium negative" and "strong negative" (each characteristic is crossed with each other characteristic and the corresponding binary relation can be weighted accordingly). The bottom of the house consists of objective measures. Measured values for arbitrary attributes are noted here. Also values of competitor's products can be filled in for direct comparison. After identifying and rating all requirements and technical characteristics, the body of the house is built: here, relationships between requirements and technical characteristics are determined. Technical characteristics can affect requirements positively or negatively. Again, relationships are classified in the range of "strong positive" and "strong negative". This weighting process should be realized in a joint effort of an engineer with the customer in order to have a common understanding of the effect of engineering characteristics on customer requirements.

Based on this concept of the HoQ we applied some changes to enable the evaluation of different product configurations from variability modeling regarding the fulfillment of customer requirements: the requirements and the importance

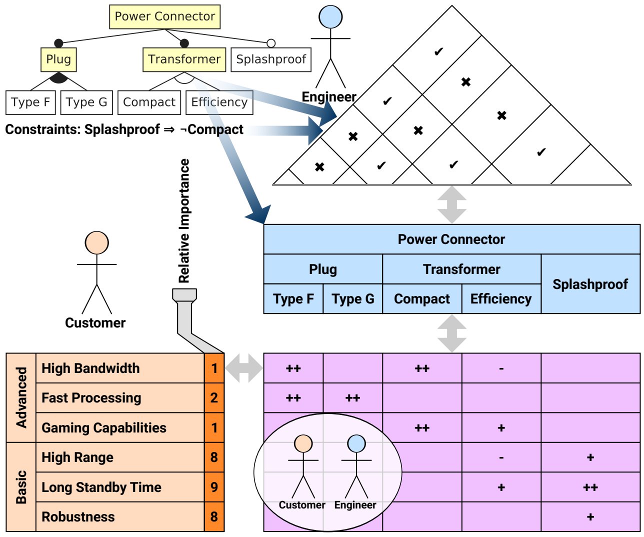

Figure 2: Refurbishing the roof of the HoQ: a feature model represents the engineering characteristics that are interconnected by cross-tree constraints and the implicit constraints resulting from the tree FM structure.

rating by the customer (i.e. the lower left part) is left unchanged. But instead of collecting engineering characteristics in the traditional way, a feature model is used for the columns and the roof of the house. The traditional way is the identification of characteristics by the engineers and finding interrelations. FM enable modeling of more complex effects. Figure 2 depicts how the adapted HoQ is built: the requirements gathered in requirements engineering are depicted as rows at the bottom left (e.g. "High Bandwidth"). Requirements prioritization is a sub-process of requirements engineering—the gathered requirements are prioritized with values in the interval of [1..9] as shown in the "chimney" column. The technical representation of the product is now realized by a feature model—the set of selectable features are the bottom line cells (e.g. "Compact", "Splashproof") of the table on top of the larger mapping matrix. That table represents the hierarchical structure of the feature model, without revealing information about the kind of edges that are used to connect features to their parents. The feature model's cross-tree constraints (as well as the constraints defined by the *tree structure* of the feature model) replace the original roof of the house. The mapping between features and requirements is shown in the larger bottom right matrix. We define a numerical mapping of relationships as shown in Table 1. The complete house (as visualized in Figure 2) is the basis for the evaluation of single products.

Table 1: Numerical mapping of relationships.

Symbol	Value	Meaning
++	2	Strong positive
+	1	Medium positive
	0	Neutral
-	-1	Medium negative
- -	-2	Strong negative

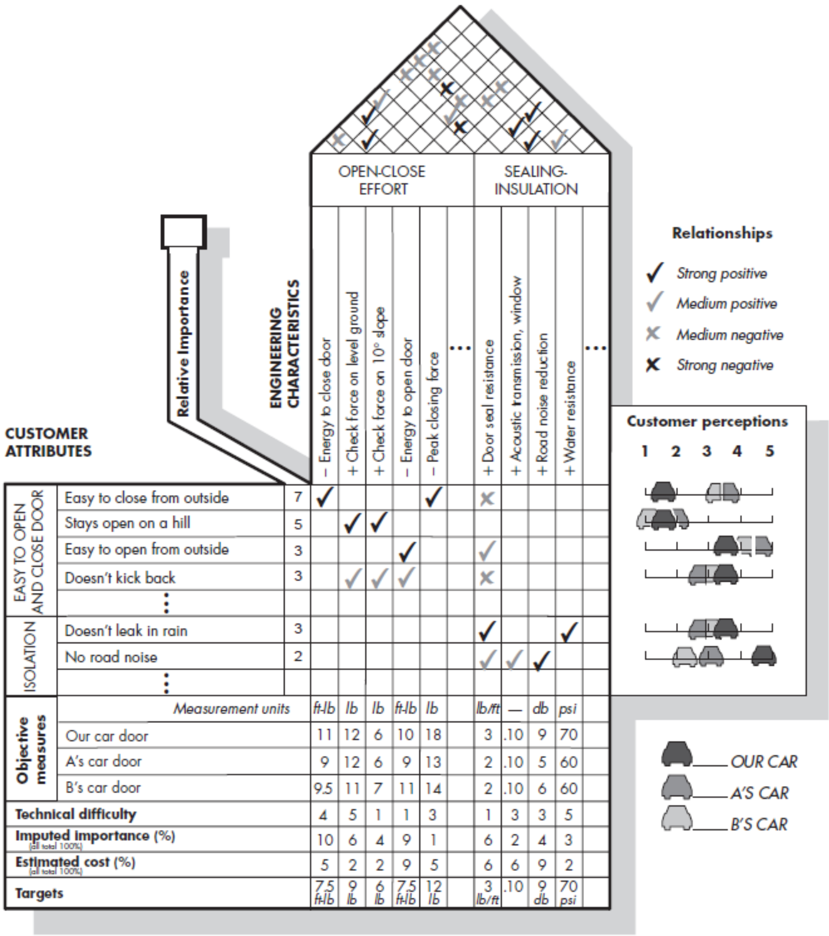

Figure 1: The original House of Quality, from [11].

3.1 Evaluation of one product

The products of interest are derived by configuring the feature model. The result is a full configuration. The selected features of a configuration stay in the configured relationship matrix, deselected features are omitted. Therefore the columns of a configured relationship matrix are a subset of the columns of the complete relationship matrix. Based on the configured relationship matrix we evaluate the configuration. Figure 3 depicts an example-configuration for a mobile phone. Based on the feature selection, every requirement is evaluated. For each requirement (row) the multiplications of the relative importance with the relationship rating are summed up. The higher the value, the better the single requirement is fulfilled by this feature selection. For comparison purposes the numeric values of all requirements are linearly mapped to grades 1 to 5, where 1 is the best and 5 is the worst level of fulfillment. Furthermore the grades are mapped to colors between green and red, where green means the best level of requirements fulfillment. Like for the requirements, the same calculation is done for each feature. For a feature a high numeric value means a strong positive effect of this feature on the fulfillment of the customer desires. A negative value means a negative effect of a single feature in terms of not meeting the requirements. By summing up the requirement-evaluations (rows) or the feature-evaluations, a scalar rating for the whole configuration can be determined.

3.2 Comparison of products

Having different products configured, the calculated value can be used for comparison purposes. A comparison between the mobile phone from Figure 3 and another, more sophisticated phone, can be seen in Figure 4. To ensure comparability, the requirements and relative importance stay the same. The two different configurations "Configuration 1" and "Configuration 2" can be seen in the columns. Each configuration has 2 columns. The values calculated in the configured relationship matrix are filled in the first column. In the second one, there is a linear mapping to values between 0% and 100% and a color mapping between green and red. The input for this mapping are not just the values of one configuration, but the values of all configurations selected for the comparison. This can be seen, as in Configuration 1 the requirement "Robustness" has a fulfillment level of 32 points, which is equal 100%. The lowest value is -36 for the requirement "Long standby time" in Configuration 2 which is equal to 0%. By summing up the values, different configurations can be compared directly. In our example Configuration 1 has 69 points and therefore a better evaluation than Configuration 2 with 25 points. This means,

	Requirement	Relative Importance	Splashproof	Low Resolution	Hard Key	High Capacity	12 Pin	Simple	GSM	BlueTooth	Compass	Europe	German			
Basic requirements	High range	8	-1						1					0	4	20%
	Long standby time	9				2		1		-1				18	2	65%
	Long conversation time	9				2								18	2	65%
	Fast charging time	4												0	4	20%
	Robustness	8	2		2									32	1	100%
	Lightness	7	-1											-7	5	3%
	Bright display	4												0	4	20%
	High contrast display	3												0	4	20%
	Works in atmosphere with high humidity	8	2											16	3	60%
	Fits into pocket well	8	-1											-8	5	0%
	Standardized connectors	1					1							1	4	23%
	Multi language support	2												0	4	20%
	Low heat development	1						1						1	4	23%
Advanced req.	High bandwith	1												0	4	20%
	High positioning precision	1									1			1	4	23%
	Fast processing	2						-1						-2	4	15%
	Fast graphics processing	1												0	4	20%
	Coupling with other devices	1								2				2	4	25%
	Gaming capabilities	1		-1	-1			-1						-3	5	13%
		79	9	-1	15	36	1	7	8	-7	1	0	0	69		
			4	4	3	1	4	4	4	5	4	4	4			
			37%	14%	51%	100%	19%	33%	35%	0%	19%	16%	16%			

Figure 3: Example evaluation of a mobile phone configuration.

that Configuration 1 meets the customer requirements better than Configuration 2 and therefore is the better choice, even if Configuration 2 is more sophisticated from a technical view.

	Requirement	Relative Importance	Configuration 1		Configuration 2	
Basic requirements	High range	8	0	53%	16	76%
	Long standby time	9	18	79%	-36	0%
	Long conversation time	9	18	79%	9	66%
	Fast charging time	4	0	53%	12	71%
	Robustness	8	32	100%	0	53%
	Lightness	7	-7	43%	0	53%
	Bright display	4	0	53%	4	59%
	High contrast display	3	0	53%	3	57%
	Works in atmosphere with high humidity	8	16	76%	0	53%
	Fits into pocket well	8	-8	41%	0	53%
	Standardized connectors	1	1	54%	1	54%
	Multi language support	2	0	53%	0	53%
	Low heat development	1	1	54%	-2	50%
Advanced req.	High bandwith	1	0	53%	2	56%
	High positioning precision	1	1	54%	4	59%
	Fast processing	2	-2	50%	4	59%
	Fast graphics processing	1	0	53%	1	54%
	Coupling with other devices	1	2	56%	4	59%
	Gaming capabilities	1	-3	49%	3	57%
	Sum		69		25	

Figure 4: Comparison between different configurations, showing that configuration 1 is superior to configuration 2 in the context of a specific customer requirements rating (relative importance).

4. CONCLUSION AND FUTURE WORK

By combining the HoQ and feature modeling we showed a way to evaluate products concerning the level on which they meet customer requirements. We showed first by an illustrative example, how a single configuration can be analyzed. This method makes it possible to understand the effects of selecting features on meeting customer requirements. Furthermore, the approach described in this work is a way to compare different product variants directly. This is especially interesting when it comes to planning a product portfolio (e.g. which variants should be offered to certain market segments). Another scenario is the evaluation of already existing products for different markets. This can easily be done by changing the relative importance of customer requirements, yielding different evaluation results.

For future work further metrics and calculations could be developed: the calculations shown here are very simple in order to introduce the proposed concept. Another issue would be to take into account effects of feature combinations: in our example a single feature has a specific (maybe none) effect on the given requirements. However, there might be scenarios where the combination of selected features can have a very different effect on specific requirements, than just the combination of each feature's single effect.

The evaluation of products as proposed in this work is only based on the level of requirements' fulfillment. The objective measures at the bottom of the house are not considered here. Since in our approach the columns of the house are the features of the product, the objective measures can be seen as the features' attributes. Clafer supports multi objective optimization based on attributed features and visualization with support for product comparison [2][1][15]. Bringing together multi objective optimization with optimization based on customer requirements could be investigated in future work.

5. ACKNOWLEDGMENTS

This work was supported by the Austrian Research Promotion Agency (FFG) within the project "Meta-Information in Model-based Variability Management"—a joint research effort of the Vienna University of Technology and SparxSystems Software GmbH.

6. REFERENCES

[1] M. Antkiewicz, K. Bąk, A. Murashkin, R. Olaechea, J. H. J. Liang, and K. Czarnecki. Clafer tools for product line engineering. In *Proceedings of the 17th International Software Product Line Conference Co-located Workshops*, SPLC '13 Workshops, pages 130–135, New York, NY, USA, 2013. ACM.

[2] K. Bąk, K. Czarnecki, and A. Wąsowski. Feature and meta-models in Clafer: Mixed, specialized, and coupled. In B. Malloy, S. Staab, and M. van den Brand, editors, *Proceedings of the Third International Conference on Software Language Engineering (SLE 2010)*, volume 6563 of *Lecture Notes in Computer Science*, pages 102–122, Berlin Heidelberg, October 2010. Springer.

[3] D. Batory. Feature models, grammars, and propositional formulas. In H. Obbink and K. Pohl, editors, *9th International Conference on Software Product Lines (SPLC 2005)*, volume 3714 of *Lecture Notes in Computer Science*, pages 7–20, Berlin Heidelberg, 2005. Springer.

[4] D. Benavides, S. Trujillo, and P. Trinidad. On the modularization of feature models. In *First European Workshop on Model Transformation (EWMT 2005), colocated with 9th International Conference on Software Product Lines (SPLC-Europe 2005)*, September 2005.

[5] T. Berger, R. Rublack, D. Nair, J. M. Atlee, M. Becker, K. Czarnecki, and A. Wąsowski. A survey of variability modeling in industrial practice. In *Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems*, VaMoS '13, pages 7:1–7:8, New York, NY, USA, 2013. ACM.

[6] K. Czarnecki, T. Bednasch, P. Unger, and U. Eisenecker. Generative programming for embedded software: An industrial experience report. In D. Batory, C. Consel, and W. Taha, editors, *Proceedings of the First ACM SIGPLAN/SIGSOFT International Conference on Generative Programming and Component Engineering (GPCE'02)*, volume 2487 of *Lecture Notes in Computer Science*, pages 156–172, Berlin Heidelberg, 2002. Springer.

[7] K. Czarnecki, S. Helsen, and U. Eisenecker. Staged configuration using feature models. In R. L. Nord, editor, *Proceedings of the Third International Conference on Software Product Lines (SPLC 2004)*, volume 3154 of *Lecture Notes in Computer Science*, pages 266–283, Berlin Heidelberg, 2004. Springer.

[8] K. Czarnecki, S. Helsen, and U. Eisenecker. Formalizing cardinality-based feature models and their specialization. *Software Process: Improvement and Practice*, 10(1):7–29, 2005.

[9] K. Czarnecki and A. Wasowski. Feature diagrams and logics: There and back again. In *Proceedings of the 11th International Conference on Software Product Lines (SPLC 2007)*, pages 23–34. IEEE, September 2007.

[10] M. L. Griss, J. Favaro, and M. d'Alessandro. Integrating feature modeling with the RSEB. In *Proceedings of the Fifth International Conference on Software Reuse*, pages 76–85. IEEE, 1998.

[11] J. R. Hauser and D. Clausing. The house of quality. *Harvard Business Review*, May–June 1988.

[12] K. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh. FORM: A feature-oriented reuse method with domain-specific reference architectures. *Annals of Software Engineering*, 5(1):143–168, 1998.

[13] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. Feature-oriented domain analysis (FODA) feasibility study. Technical Report CMU/SEI-90-TR-21, DTIC Document, November 1990.

[14] K. C. Kang, J. Lee, and P. Donohoe. Feature-oriented product line engineering. *IEEE Software*, 19(4):58–65, 2002.

[15] A. Murashkin, M. Antkiewicz, D. Rayside, and K. Czarnecki. Visualization and exploration of optimal variants in product line engineering. In *Proceedings of the 17th International Software Product Line Conference*, SPLC '13, pages 111–115, New York, NY, USA, 2013. ACM.

[16] R. Olaechea, S. Stewart, K. Czarnecki, and D. Rayside. Modelling and multi-objective optimization of quality attributes in variability-rich software. In *Proceedings of the Fourth International Workshop on Nonfunctional System Properties in Domain Specific Modeling Languages*, page 2. ACM, 2012.

[17] M.-O. Reiser. *Managing Complex Variability in Automotive Software Product Lines with Subscoping and Configuration Links*. PhD thesis, Technische Universität Berlin, December 2008.

[18] M. Riebisch, K. Böllert, D. Streitferdt, and I. Philippow. Extending feature diagrams with UML multiplicities. In *Proceedings of the Sixth Conference on Integrated Design and Process Technology (IDPT 2002), Pasadena, CA*, 2002.