USING MDM-METHODS IN ORDER TO IMPROVE MANAGING OF ITERATIONS IN DESIGN PROCESSES

Sebastian Kortler1, Bergen Helms1, Marina Berkovich2, Udo Lindemann1, Kristina Shea1, Jan Marco Leimeister3 and Helmut Krcmar2

1Institute of Product Development, Technische Universität München, Germany
2Chair for Information Systems, Technische Universität München, Germany
3Chair for Information Systems, Universität Kassel, Germany

Keywords: MDM, requirements engineering, cycles in innovation processes, design processes

1 INTRODUCTION
The complexity of products, development processes and organizational structures is constantly increasing. The reasons therefore are manifold, like: increasing complexity of the customers’ requirements regarding the quality; increasing pressure of time and cost in the manufacturing and development processes; integration of multiple domains within one product, like mechanics, software, electronics or service (Lindemann, 2006; Pulm, 2004; Leimeister and Glauner, 2008; Abramovici and Schulte, 2007). Consequently, these challenges force the companies to enhance their innovation capabilities and to cope with the increasing complexity.

In development processes, the requirements engineering (RE) plays a vital role: It is “the process of discovering the purpose of the system-to-be, by identifying stakeholders and their needs, and documenting these in a form that is suitable to analysis, communication, and subsequent implementation” (Nuseibeh and Easterbrook, 2000) RE is. In order to efficiently translate requirements into properties of the product, it is necessary to integrate the RE with the subsequent steps of the product development process. However, requirements’ changes are often discovered only in the late phases of the development (Lindemann and Reichwald, 1998) and this results into a costly modification of all affected parts of the system. In order to react on changes efficiently, it is necessary to develop the requirements and the design in parallel and in a coordinated manner.

Up to now, there is no systematic and formal process on how requirements can be transformed into product properties. The particular challenge in this process lies in the iterations between the ongoing refinement of requirements and the design of the properties of the future product. The research presented in this paper aims to tackle this challenge with a formal procedural method that builds on matrix-based methods. Based on the interpretation of structural criteria, this approach guides that iterative process.

This paper is structured as follows: After the presentation of the background and related work, the procedural model for guiding the iterations between requirements engineering and the detailing of the design artifact is detailed in section 3. Next, an exemplary validation of the method is given in section 4. Finally, the paper closes with an outlook and a conclusion.

2 BACKGROUND AND RELATED WORK
The aim of requirements engineering is to elicit, analyze and document requirements to the design problem to be solved (Sommerville, 2005, Humpert, 1995). In software engineering, the requirements analysis is defined as a concretization of initial requirements to the product-to-be. During this task the requirements are checked for consistency and completeness (Cheng and Atlee, 2007, Sommerville and Kotonya, 1998). The resulting requirements are solution-independent and formulated in a fashion that is comprehensible to the developer. Also in product development, requirements are translated into the language of the developers and checked for inconsistencies (Lindemann, 2006, Pahl et al., 2006). Unlike in software engineering, in product development RE is not understood as an independent discipline. It is rather described as a part of the general development process. Another important task
of RE covered in both disciplines is the change management, which includes the impact analysis of changing requirements (Sommerville and Kotonya, 1998, Lindemann, 2006). If requirements are changed their interdependencies and potential change propagation have to be considered and handled in an adequate way. In order to fulfill these tasks of change management, it is essential to model the dependencies between requirements and development artifacts. The change management is often not explicitly handled by the RE. Only the advice is given, to collect changes and apply them in iterations, whereby the updating of a central repository of requirements is important (Cheng and Atlee, 2007, Sommerville and Kotonya, 1998, Pahl et al., 2006).

The shift of perspective from a user-oriented requirements model to a product-centric description of the design artifact is realized in the conceptual phase of the product development process. This phase is methodologically well supported in the classic design literature (Pahl et al., 2006). In order to bridge the gap between high-level requirements and a concrete product architecture, it is a common approach to establish a functional model based on the description of requirements. This model is transferred – via a physical but idealized representation – to a product architecture that is suitable for accomplishing the behaviors and thus the required functionality with respect to the requirements. This process is highly iterative, e.g. due to the idealized nature of physical effects that induce unintended behaviors when being embodied in components. Furthermore, as the technological knowledge about the future product is growing throughout the development process, an ongoing refinement respectively adjustment of the product requirements is necessary.

Depending on the formulation of the design problem, suitable modeling techniques have to be chosen. While relation-oriented function models capture logical dependencies between function blocks, e.g. for the fulfillment of Function 1 a prior Function 2 is required. Flow-oriented function models represent the interplay of functions based on the exchange of, e.g., energy, material or signals. Most of the modeling techniques in this area, also on the behavioral and structural level, are based on a block-oriented representation and provide thus a foundation for being formalized in a graph-based representation.

For an integrated modeling of requirements and product characteristics, modeling approaches are needed that are able to capture multiple phases but also multiple domains due to the interdisciplinary nature of product development processes. A promising approach, stemming from Systems Engineering, is the modeling language SysML. SysML provides a modeling framework for an integrated consideration of both multiple domains and multiple phases (Wölk and Shea, 2009). Nevertheless, it does not provide procedural support for the iterative refinement and modeling of requirements, functions, physical effects and product architectures.

Co-design is another approach that rather focuses on procedural aspects (Pohl and Sikora, 2007). It has the advantages that the requirements can be concretized early in the development process. Moreover, the appearance of further requirements during the design phases will not automatically lead to huge changes in the requirements and product models. In the first steps of the co-design, the product is structured into functions describing the tasks the product is expected to perform. This concretization needs multiple iterations, in which the functions are defined and the requirements are assigned to functions. However, this approach does not give indications where requirements have to be refined or properties of the design artifact have to be further detailed. Especially, when the modeling involves different levels of abstraction the dependencies between design properties and requirements become increasingly complex and are thus difficult to address. The approach presented in the following sections responds to that need. It uses DSM techniques in order to control the iterative refinement of design properties and requirements.

3 PROCEDURAL MODEL FOR THE REFINEMENT OF REQUIREMENTS AND FUNCTIONS

This research claims its applicability throughout the entire iterative process of refining requirements and concretizing product properties. “While the reference architecture increasingly takes shape, the need for more detailed requirements arises as well” (Pohl et al., 2005). The presented approach supports to reduce complexity of the design challenge by stepwise connecting the requirements with the design. Therefore, the authors present a procedural model (depicted in figure 1). The model contains two main parts: Requirement modeling and conceptual design. The refinement tasks of the requirements and of the conceptual design are performed in parallel. Using MDM-analysis after each
refining iteration and mapping of the models will support the next refining step of both models. To do so, the MDM-analysis indicates which function and which requirement need to be refined within which iteration as described subsequently.

The model includes 5 iterative process steps depicted in figure 1. First of all, market and customer needs are elicited and translated to requirements (step 1). Afterwards, the requirement modeling structures the collected requirements into different groups (step 2). In step 3, the conceptual design is performed. In order to fulfill the requirements of the requirement model, a functional model is created. After mapping the requirements model on the functional model (step 4), matrix-based methods are used to analyze relations between requirements and functions (step 5). Therefore, the proposed method captures requirements and functions and their relations inside inter domain matrices which allow for modeling elements of two separated domains, figure 2. Moreover, the structure of the inter domain matrix is analyzed with the help of structural criteria. Assumption of case a: The higher the number of requirements a function F is fulfilling, the more abstract the function F is expressed. Thus, function F has a high potential of refinement. Assumption of case b: The higher the number of functions a requirement R needs to be fulfilled, the more abstract the requirement R is expressed. Thus, requirement R has a high potential of refinement. More precisely, the active sum of the respective modeled function indicates the potential of refinement (case a), whereas the passive sum of the respective modeled requirements indicates the potential of refinement (case b). Performing the analysis will lead to the following cases:

a. Functions with a high active sum: Potential of refinement or concretization of the respective function
b. Requirements with a high passive sum: Potential of refinement or concretization of the respective requirement
c. Functions with active sum = 0: Indications of useless functions or missing requirements
d. Requirements with passive sum = 0: Indication of missing functions (failed requirement)

In this way the matrix-based analysis provides the possibility of controlling the refinement processes in each step of iteration. During the development process new requirements may arise. On the one
hand, this model allows for including requirements at any time. The matrix-based analysis will quickly
direct to failed requirements. Hence, new functions fulfilling added requirements can be included. On
the other hand, useless functions can be detected and removed afterwards with the proposed model as
well. The functional model just as the requirement model will be improved during the continuous
development process. Finally, the modeling processes of requirements and functions are accomplished
if there is none or low potential of concretization and all requirements are fulfilled.

Our approach supports the variant management meaning. We consider the functions as modules with
specific functionalities and being adaptable easily (Beverungen et al., 2008). These modules can be
standardized and reused in other products.

4  EXAMPLE: LAUNDRY ON DEMAND

In order to confirm the presented method, the authors present iterations between RE and conceptual
modeling. This example explores systematically the design of a product service system (PSS) that
provides the customer clean laundry on demand. This example is targeted towards an application in
the commercial sector.

The first iteration, figure 3, is initialized based on the investigation of market and customer needs so
that requirements can be modeled. These can be grouped into three categories: cleaning laundry,
scheduling of cleaning and invoicing. On this foundation a functional model is created. In the fourth
step an inter domain matrix is established that captures requirements, functions and their relations.
Afterwards, the matrix-based analysis is performed. The results indicate a high potential of refinement
concerning function 2: Washing laundry and requirement 1: Clean laundry needs to be returned to the
customer. Finally, the first iteration of the proposed method is completed.

![Figure 3. First Iteration: requirements, functions and model of their mapping](image)

In order to perform the second iteration of the proposed method, function 2 and requirement 1 are
refined. Thus, the requirements model as well as the functional model is concretized and the iterative
method can be used again (see figure 4). These refinement iterations can be applied continuously
during the ongoing development process.

![Figure 4. Iteration 2, Indication of high potential of refinement (F2d, R1a, R1e)](image)

5  FUTURE WORK

Our broader research focuses on PSS, often also called hybrid products or complex solutions because
due to tight market competition, differentiation by products or services alone cannot be achieved any
more (Böhmann and Krcmar, 2007, Leimeister and Glauner, 2008). The development of complex
solutions is challenging and faces similar problems like product development. Particularly, the requirements engineering for complex solutions is challenging, because all domains involved in the development – product, software, and service engineering – have different understandings of the term “requirement” and of the RE’s tasks and activities. In literature as well as in practice, products and services are developed separately (Leimeister and Glauner, 2008). For the area of requirements engineering, the challenge for research and practice is to develop methods for the translation of requirements into “the language of the developer” and partition them according to the domains involved.

In our future research we will apply the presented method of refining requirements and functions to complex solutions. By the use of this method, the complex solution is structured by means of functions and all requirements are assigned to those functions. Then, each function is assigned to a dedicated domain, which has to realize it. That means each function is realized by either product-, software-, or service engineering. This method will be part of our integrated requirements engineering approach for complex solutions. Such an integrated approach will enable a continuous managing of requirements throughout the development process.

Moreover, the proposed model uses very simple structural criteria. In future work, the authors will analyze the structure of requirements. Therefore, the structure of relation-oriented functional models and function-requirement-matrices is transferred to requirements DSMs. These DSMs are applicable for further structural criteria e.g. cluster analysis. Furthermore, the authors will identify respective dependencies in order to illustrate possible change propagation.

Previous and ongoing work in the field of computational design synthesis will be combined with this research (Helms et al., 2009). Consequently, graph grammars will be integrated in the presented procedural model and provide the methodological support for automating (partially) the design. This requires a formal modeling representation, like the Functional Basis (Hirtz et al., 2002). Furthermore, the inclusion of modeling elements from design catalogues will be supported.

6 CONCLUSION
The development and manufacturing times are decreasing due to the raising competition. Nevertheless, the customers’ requirements become more complex regarding the quality of products and services. In order to react to changes of requirements efficiently, it is necessary to develop the requirements and the design in parallel. This paper proposes an approach for controlling and reducing iterations between requirements engineering and functional modeling using matrix-based methods. More precisely, the proposed model indicates which function and which requirement need to be refined within which iteration. To do so, the requirements model is mapped on the functional model using inter domain matrices. All functions and requirements as well as their relations are captured within each step of iteration. In order to identify functions and requirements with a high potential of refinement, the authors used active and passive sum considering the inter domain matrices. Thereby, missing function, useless functions, and missing requirements can be identified as well. The application of the iterative refinement model is presented using an example concerning a laundry on demand PSS.

In future work, the authors will concentrate on extending the proposed method, using further structural criteria in order to analyze structural dependencies between requirements through functions. Moreover, the authors extend their integrated modeling approach and apply the presented method of refining requirements and functions to complex solutions.

ACKNOWLEDGEMENT
We thank the Deutsche Forschungsgesellschaft (DFG) for funding this research as a part of the collaborative research centre “Managing cycles in innovation processes – Integrated development of product service systems based on technical products” (SFB 768).

REFERENCES


Using MDM-Methods in Order to Improve Managing of Iterations in Design Processes

Sebastian Kortler¹, Bergen Helms¹, Marina Berkovich²,
Udo Lindemann¹, Kristina Shea¹,
Jan Marco Leimeister³ and Helmut Krcmar²
¹Institute of Product Development, Technische Universität München
²Chair for Information Systems, Technische Universität München
³Chair for Information Systems, Universität Kassel

Index

• Background and motivation
• Procedural model for the refinement of requirements and functions
• Example: Laundry on Demand
• Future work
**Background and motivation**

- The complexity of products, development processes and organizational structures is constantly increasing
- In development processes, the requirements engineering (RE) plays a vital role
- Requirements’ changes are often discovered only in the late phases of the development – this results into a costly modification of all affected parts of the system
- In order to fulfill these tasks of change management, it is essential to model the dependencies between requirements and development artifacts
- In order to bridge the gap between high-level requirements and a concrete product architecture, it is a common approach to establish a functional model based on the description of requirements
- This model is transferred – via a physical but idealized representation – to a product architecture

---

### Lists of requirements

1. Statutory requirements
   - 031: Laundry of hotel guests need to be mapped to guests room number
   - 050: Before removing laundry from inventory the amount need to be reported
   - 002: Cleaned laundry need to be delivered to the customer immediately
   - 060: The cleaning process must not exceed 10 liter water / kg

2. Performance oriented requirements
   - 051: Bedclothes need to be stored in the inventory before 9:00 am
   - 055: Before removing laundry from inventory the guest’s room number need to be reported
   - 047: Laundry need to be stored in the inventory after 9:00 am
   - 030: Tableclothes need to be blanched after cleaning
   - 060: The cleaning process must not exceed 10 liter water / kg
   - 002: Cleaned laundry need to be delivered to the customer immediately
   - 051: Bedclothes need to be stored in the inventory before 9:00 am

---

### Functional model

- **Adding function**
- **Changing requirement**
- **Adding requirement**
- **Refinement and Concretization**
BY MODELING DEPENDENCIES  MANAGING COMPLEXITY

Procedural model for the refinement of requirements and functions

1. Market and customer needs
2. Requirements modeling
3. Conceptual design
4. Mapping requirements on functions
5. Matrix-based analysis

Case a
Case b
Case c
Case d

Update

Adding requirement
Refinement / Concretization

Iterations

Removing function
Adding function

Product System Service (PSS) Laundry on Demand

Hotel

Solving problem, defining general conditions

Solving problem

Solution provider

Problem: Clean Laundry

Clean Laundry

Ironing

Cleaning

Transport
### Using market and customer needs for requirement modeling (1st Iteration)

#### Lists of requirements

<table>
<thead>
<tr>
<th>statutory requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. statutory requirements</td>
</tr>
<tr>
<td>2. performance oriented requirements</td>
</tr>
<tr>
<td>3. process oriented requirements</td>
</tr>
<tr>
<td>031: Laundry of hotel guests need to be mapped to guest's room number</td>
</tr>
<tr>
<td>055: Before removing laundry from inventory the amount need to be reported</td>
</tr>
<tr>
<td>002: Cleaned laundry need to be delivered to the customer immediately</td>
</tr>
<tr>
<td>060: The cleaning process must not exceed 10 liter water / kg</td>
</tr>
<tr>
<td>030: Table clothes need to be blanched after cleaning</td>
</tr>
<tr>
<td>047: Laundry need to be stored in the inventory after sorting</td>
</tr>
<tr>
<td>051: Bedclothes need to be stored in the inventory before 9:00 am</td>
</tr>
<tr>
<td>060: The cleaning process must not exceed 10 liter water / kg</td>
</tr>
<tr>
<td>030: Table clothes need to be blanched after cleaning</td>
</tr>
<tr>
<td>047: Laundry need to be stored in the inventory after sorting</td>
</tr>
<tr>
<td>051: Bedclothes need to be stored in the inventory before 9:00 am</td>
</tr>
<tr>
<td>002: Cleaned laundry need to be delivered to the customer immediately</td>
</tr>
</tbody>
</table>

#### Conceptual design (1st Iteration)

**Functional model**

- Cleaning laundry
- Planning Process
- Bring to account

**Tasks:**
- F1: Collecting laundry
- F2: Cleaning laundry
- F3: Drying laundry
- F4: Ironing laundry
- F5: Sorting laundry
- F6: Delivering laundry
- F7: Get information
- F8: Planning delivery time
- F9: Planning process steps
- F10: Calculation of the amount of laundry
- F11: Generating monthly report
- F12: Generating bill
- F13: Delivering bill

**Caption:** Function F1 is required of Function F2
Mapping requirements on functions (1st Iteration)

- Domain-Mapping-Matrix (DMM) models functions fulfilling requirements

- DMM uses functional model
- DMM uses requirement model

Matrix-based analysis (1st Iteration)

MDM-analysis after each refining iteration and mapping of the models

The higher the number of requirements a function F is fulfilling, the more abstract the function F is expressed.

The higher the number of functions a requirement R needs to be fulfilled, the more abstract the requirement R is expressed.

- Case a: Functions with a high active sum: Potential of refinement or concretization of the respective function
- Case b: Requirements with a high passive sum: Potential of refinement or concretization of the respective requirement
- Case c: Functions with active sum = 0: Indication of useless functions or missing requirements
- Case d: Requirements with passive sum = 0: Indication of missing functions
Refinement of the function “cleaning laundry”

Refinement of F2

Lists of requirements

1. Statutory requirements
2. Performance oriented requirements
3. Process oriented requirements

Lists of requirements

1. Statutory requirements
2. Performance oriented requirements
3. Process oriented requirements

Refinement of the requirement “Cleaned laundry needs to be returned to the customer”

Refinement of Requirement R1

1. Statutory requirements
2. Performance oriented requirements
3. Process oriented requirements

- R1a: Laundry need to be delivered before 10 pm
- R1b: Bedclothes need to be delivered before 9 pm
- R1c: Laundry need to be sorted to the inventory
- R1d: Blanched laundry need to be sorted separately
- R1e: Bedclothes need to be sorted separately
- R1f: The inventory must be locked after delivering cleaned laundry
### Mapping requirements on functions (2nd Iteration)

<table>
<thead>
<tr>
<th>Func-Req</th>
<th>R1a</th>
<th>R1b</th>
<th>R1c</th>
<th>R1d</th>
<th>R1e</th>
<th>R1f</th>
<th>R1g</th>
<th>R1h</th>
<th>R1i</th>
<th>R1j</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2a</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2d</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2f</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F2g</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Matrix-based analysis (2nd Iteration)

- In this way the matrix-based analysis provides the possibility of controlling the refinement processes in each step of iteration.
- During the development process new requirements may arise and this model allows for including requirements at any time.
- New functions fulfilling added requirements can be included. Useless functions can be detected and removed afterwards with the proposed model.
- The functional model just as the requirement model will be improved during the continuous development process.
- Finally, the modeling processes of requirements and functions are accomplished if there is none or low potential of concretization and all requirements are fulfilled.
Conclusion

• In order to react to changes of requirements efficiently, it is necessary to develop the requirements and the design in parallel
• This paper proposes an approach for controlling and reducing iterations between requirements engineering and functional modeling using matrix-based methods
• The proposed model indicates which function and which requirement need to be refined within which iteration
• All functions and requirements as well as their relations are captured within each step of iteration
• In order to identify functions and requirements with a high potential of refinement, the authors used active and passive sum considering the interdomain matrices
• Missing function, useless functions, and missing requirements can be identified with this method
• The application of the iterative refinement model is presented using an example concerning a laundry on demand PSS

Future work

• In our future research we will apply the presented method of refining requirements and functions to further PSS
• The authors will analyze the structure of requirements
  – The structure of relation-oriented functional models and function-requirement-matrices is transferred to requirements DSMs
  – These DSMs are applicable for further structural criteria e.g. cluster analysis
  – Furthermore, the authors will identify respective dependencies in order to illustrate possible change propagation
• Previous and ongoing work in the field of computational design synthesis will be combined with this research
  – Graph grammars will be integrated in the presented procedural model and provide the methodological support for automating (partially) the design