Task Definition for Design Automation

Author(s):
Rigger, Eugen

Publication Date:
2019

Permanent Link:
https://doi.org/10.3929/ethz-b-000382356

DISS. ETH NO. 26019

Task Definition for Design Automation

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

Eugen Karl Rigger

MSc ETH, ETH Zurich

born on 24.02.1989
citizen of Austria

accepted on the recommendation of

Prof. Dr. Kristina Shea
Prof. Dr. Alex Duffy
Dr. Tino Stanković

2019

Abstract

Design automation has been the focus of research for more than five decades. It supports design processes in several aspects by automating design tasks based on computational methods and tools to save time, generate alternative design solutions, explore solution spaces, and reuse engineering knowledge. Yet, the current industrial practice does not reflect the opportunities provided by state-of-the-art design automation methods. The factors contributing to this gap are: first, a lack of knowledge of design automation opportunities and insufficient support for the integration of design automation in design practice including the supporting methods and technological environments. Second, metrics and methods for comprehensive estimation of the impact of design automation implementation on design practice do not exist making it difficult to quantify the value of design automation and justify efforts for implementation. Finally, design automation applications are often perceived as black-box systems since knowledge is hard-coded in design automation applications. This also increases efforts for knowledge formalization.

In response to these issues, this thesis proposes a methodology for design automation task definition that features collaborative workshops to account for the different viewpoints of designers. It builds upon a design automation task categorization that is characterized by the knowledge levels required for design automation task definition and consists of four different methods that build on each other. The first method focuses on the identification of design automation use cases. It features detailed analysis of design processes and reuse of design automation task templates to support both the identification of possible use cases and the integration of the corresponding software applications into design practice. The second method introduces a top-down derivation of metrics based on potential failure modes in design. The third method enables estimation of the impact and value of design automation implementation based on design automation task templates enabling reuse and associating metrics to design processes. Finally, design automation task formalization by designers is enabled using graphical modeling. The method supports reuse and modularization of knowledge based on the design automation task categorization. To enable reasoning in the context of the methodology, a meta-model that clarifies the vocabulary is established based on standardized languages.

The proposed methodology is evaluated based on three industrial use cases that highlight the necessity to involve multiple designers for design automation task definition to account for different viewpoints for needs identification. Further, the results show the potential for design automation application in the early stages of design as well as the applicability of the proposed approach for design automation task formalization by designers. Thus, the work presented in this thesis contributes by introducing and evaluating a novel methodology for design automation task definition that brings the opportunities of state-of-the-art design automation methods into line with designers' needs. It extends the state-of-the-art with respect to new methods supporting design automation task definition and also consolidates research in design automation. Thereby, the methodology enables alignment of research on design automation and increases awareness of design automation opportunities in industry among the different stages of the design process.

Zusammenfassung

Die Automatisierung des Entwurfsprozesses – Design Automation – ist bereits seit mehr als 5 Jahrzehnten Gegenstand der Forschung. Die Automatisierung von Aufgaben des Entwurfsprozesses durch formalisiertes Wissen, Algorithmik und zugehöriger Software ermöglicht Zeitersparnisse, das Untersuchen von Lösungsräumen und Generieren von Lösungsalternativen sowie das systematische Wiederverwenden von Ingenieurswissen. Allerdings stehen sich State-of-the-Art und State-of-Practice im Ingenieurswesen teilweise diametral gegenüber. Einige Gründe dafür sind, erstens, das fehlende Wissen hinsichtlich der Möglichkeiten von Design Automation, sowie die mangelnde Integration von Design Automation Lösungen in die Ingenieurspraxis. Dabei gilt es sowohl die Schnittstellen des Entwurfsprozesses, als auch die der zugehörigen Methoden und Technologien fallspezifisch zu berücksichtigen. Zweitens, gibt es einen Mangel an Methoden sowie zugehöriger Messgrößen zur quantitativen Abschätzung des Nutzens von Design Automation in der Ingenieurspraxis. Dadurch können die notwendigen Kosten für die Implementierung nur schwer gerechtfertigt werden. Nicht zuletzt werden Design Automation Lösungen als Black-Box Systeme empfunden, da das Ingenieurswissen in der Software hardcodiert und für den Anwender nicht zugänglich implementiert wird. Außerdem muss dafür das Wissen von Ingenieuren erfasst und formalisiert werden, was den Aufwand für die Implementierung einer Design Automation Lösung steigert.

Bezugnehmend auf diese Punkte wird in dieser Arbeit eine vierstufige Methodik zur Definition von Design Automation Aufgaben in spezifischen Ingenieurskontexten vorgestellt. Die Methodik zeichnet sich durch eine kollaborative Vorgehensweise mit Ingenieuren aus, um so den unterschiedlichen Standpunkten gerecht zu werden und die Anforderungen umfassend zu erfassen. Die Vorgehensweise baut auf einer Design Automation Aufgabenkategorisierung auf, welche die Aufgabentypen basierend auf den Möglichkeiten des State-of-the-Art und dem für die Formalisierung notwendigen Wissen beschreibt. Dementsprechend befasst sich der erste Schritt der Methodik mit der Identifikation möglicher Use Cases für die Anwendung von Design Automation in der Praxis. Eine detaillierte Analyse des Entwurfsprozess sowie die Verwendung von Design Automation Aufgabentemplates ermöglichen es Anwendungsfälle zu identifizieren und diese auch entsprechend in die Produktentwicklungsumgebung zu integrieren. Der zweite Schritt befasst sich mit der Ableitung von Messgrößen basierend auf möglichen Schwachstellen im Entwurfsprozess. Im dritten Schritt werden diese Messgrößen verwendet um den Einfluss der Implementierung von Design Automation auf die Praxis quantitativ abzuschätzen. Nach positiver Bewertung einer Design Automation Lösung zielt der letzte Schritt der Methodik auf die eigenständige Formalisierung des Wissens durch die Ingenieure ab. Formalisierung, Modularisierung und Wiederverwendung des Wissens werden durch die Anwendung von Templates sowie einer graphischen Modellierungssprache ermöglicht. Die Methodik wendet ausschließlich graphische, standardisierte, formale Sprachen an und basiert auf einem Meta-Modell, welches das angewendete Vokabular begründet sowie die semantischen Zusammenhänge erläutert.

Die Validierung der Methodik wurde anhand von drei industriellen Anwendungsfällen durchgeführt. Dabei wurde die Notwendigkeit der kollaborativen Vorgehensweise für die umfassende Erfassung der Anforderungen für Design Automation festgestellt. Außerdem wurde das Potential für die Anwendung von Design Automation in den frühen Phasen des Entwurfsprozesses erstmals systematisch festgehalten, sowie das Potential für die eigenständige Wissensformalisierung durch die Ingenieure aufgezeigt. Daher liegt der Beitrag dieser Arbeit in einer neuen Methodik welche den Abgleich der zur Verfügung stehenden Möglichkeiten von Design Automation mit den Anforderungen der Ingenieurspraxis ermöglicht. Die einzelnen Schritte der Methodik beinhalten neue Vorgehensweisen und führen bestehende Arbeiten verschiedener Forschungsfelder aus dem Umfeld von Design Automation zusammen. Nicht zuletzt liegt der Mehrwert dieser Arbeit darin, dass die Möglichkeiten von Design Automation entlang des gesamten Entwurfsprozesses in der Praxis erkannt werden.

Acknowledgements

First, I would like to thank my supervisor Prof. Kristina Shea for providing me the opportunity to do an external PhD at EDAC and the continuous support over the years of this PhD project. I am grateful for the challenges you provided, the discussions we had and the time you invested in me.

Next, my thanks go to Dr. Tino Stanković who oftentimes encouraged me in my work through his enthusiasm for the topic design automation and good jazz music. Many thanks for the countless hours you spent in teaching me how to scientifically write papers and the late night feedback on my papers.

I would also like to thank Prof. Alex Duffy for his expertise and for examining my thesis. Thanks for introducing me to the definitions of design tasks, design activities etc. which enabled me to recognize the subtleties of the research field.

Also, I want to express my gratitude to my supervisors at V-Research: Dr. Vaheh Khachatouri for offering me the challenge to both work in industry and pursue a PhD at the same time; Dr. Heinz Seyringer for his enthusiasm and continuous trust in the quality of my work; and; Dr. Thomas Vosgien for energetic and challenging discussions that helped me to further elaborate and mature my ideas related to design automation task definition. Special thanks to Martin Schwarz for constructive feedback on my ideas and providing me the opportunity for conducting the industrial case studies.

Thanks go to the EDAC team that always welcomed me even though I was sort of a rare guest in the past 4 years. In particular, thanks go to Kathy, who provided helpful support when I was stuck in optimization problems or writing my thesis and the students Sebastien and Alexander. Similarly, a salute goes to my colleagues at V-Research and my team that supported me even within times that I was rarely accessible and potentially stressed out due to several concurrent challenges.

Personally, I want to thank my family and friends for supporting me in all facets of my life. More particularly, I want to thank my parents Ulrike and Erich for offering me the chance to pursue the education I want and encouraging me on whatever path I was pursuing.

Last but not least, my thanks go to Sabrina for her everlasting support and patience. Your positivism and restless energy enabled me to go to my limits (and potentially beyond).

This work was partially supported and funded by the K-Project ‘Advanced Engineering Design Automation’ (AEDA) and the “Austrian Competence Center for Digital Production” that are financed under the COMET (COMpetence centers for Excellent Technologies) funding scheme of the Austrian Research Promotion Agency.

Contents

Abstract i

Zusammenfassung iii

Acknowledgements v

Contents vii

Acronyms xi

1 Introduction 1

1.1 Objectives and Contributions 3

1.2 Expected Contributions 3

1.3 Research Questions 4

1.4 Research Methodology 6

1.5 Structure of the Thesis 7

2 Related Work 9

2.1 Design Automation 9

2.1.1 Design Automation Terminology 9

2.1.2 Identification Design Automation Use Cases 10

2.1.3 Design Automation Integration 12

2.2 Estimation of the Impact of Design Automation on Design Practice 14

2.2.1 Methods for Impact Estimation of Design Automation 14

2.2.2 Characteristics of Design Performance Metrics 16

2.2.3 Top-down Derivation of Metrics 16

2.2.4 Drivers and Metrics 18

2.3 Design Automation Task Formalization 24

2.3.1 Design Automation Task Knowledge 24

2.3.2 Methodologies for Design Automation Task Formalization 26

2.3.3 SysML-based Design Automation Task Definition 26

2.4 Summary 27

3 Industrial Cases 31

4 Methodology for Design Automation Task Definition 33

4.1 Overview of Methodology for Design Automation Task Definition 33

4.1.1 The Design Automation Task Categorization 34

4.1.2 Step 1 – Identification of Design Automation Use Cases 35

4.1.3 Step 2 – Derivation of Metrics for Impact Estimation of Design Automation 35

4.1.4 Step 3 – Estimation of the Impact of Design Automation on Design Practice 35

4.1.5 Step 4 – Formalization of the Design Automation Task 36

4.1.6 Steps not covered in the scope of this work 36

4.2 Meta-Model for Integrated Assessment of Design Processes 37

4.3 Evaluation of Methodology for Design Automation Task Definition 38

5 Design Automation Task Categorization and Templates 39

5.1 Criteria for Systematic Analysis of Design Automation Methods with respect to Design Automation Tasks 39

5.1.1 Input / Output 40

5.1.2 Goals 42

5.1.3 Design Automation Methods 42

5.1.4 Criteria and Instance Values for Systematic Analysis 44

5.2 Design Automation Task Categorization 44

5.2.1 Analysis of State-of-the-Art Design Automation 45

5.2.2 Design Automation Task Categorization 45

5.3 Derivation of Design Automation Task Templates 51

5.3.1 Design Automation Task Templates – Process Perspective 51

5.3.2 Design Automation Task Templates – Motivational Perspective 53

5.3.3 Design Automation Task Templates – Product Knowledge Perspective 53

5.4 Discussion 54

5.4.1 Knowledge Levels for Design Automation Task Definition 54

5.4.2 Consolidation of Research Fields 56

5.5 Summary 58

6 Method for Identification of Design Automation Use Cases 61

6.1 Method 61

Workshop Organization 66

6.2 Results 66

6.2.1 Workshop Setup 67

6.2.2 Workshop Results 67

6.3 Discussion 72
6.3.1 Validation of Design Automation Use Cases 72
6.3.2 Design Automation Integration 73
6.3.3 Implications for Design Practice 74
6.3.4 Implications for Design Research 75
6.4 Summary 75
7 Method for Derivation of Metrics for Impact Estimation of Design Automation 79
7.1 Method 79
Workshop Organization 85
7.2 Results 85
7.2.1 Workshop Setup 85
7.2.2 Workshop Results 86
7.3 Discussion 91
7.3.1 Application Evaluation 91
7.3.2 Success Evaluation 92
7.3.3 Implications for Design Practice 94
7.3.4 Implications for Design Research 95
7.4 Summary 95
8 Method for Estimation of the Impact of Design Automation on Design Practice 97
8.1 Method 97
Workshop Organization 99
8.2 Results 100
8.2.1 Research Methodology 100
8.2.2 User Study Results 104
8.3 Discussion 108
8.3.1 Investigation of Rating Scales 108
8.3.2 Impact of a Design Automation Prototype 109
8.3.3 Implications on Design Practice 110
8.3.4 Implications on Design Research 111
8.4 Summary 111
9 Method for Design Automation Task Formalization 113

9.1 Method 113
Design Automation Task Definition for PATS/PAPS 114
9.2 Results 120
9.2.1 Research Methodology 120
9.2.2 User Study Results 125
9.3 Discussion 127
9.3.1 Indications of User Study and Implications on Design Practice 127
9.3.2 Comparison to related approaches 129
9.3.3 Implications on Design Research 131
9.4 Summary 132
10 Discussion 135
10.1 Applied Research Methodology 135
10.2 Implications on Design Practice 135
10.3 Implications on Design Research and Contributions 137
10.4 Overall Research Question 139
10.5 Limitations and Future Work 139
11 Summary and Conclusion 143
Bibliography 145
Appendix 167

Acronyms

BDD	SysML Block Definition Diagram
CAD	Computer Aided Design
CAx	Computer Aided Technologies
CDS	Computational Design Synthesis
DA	Design Automation
dpFMEA	Design Process Failure Mode Effects Analysis
DRM	Design Research Methodology
EA	Enterprise Architecture
GQM	Goal-Question-Metric
IBD	Internal Block Diagram
KBE	Knowledge-based engineering
KBS	Knowledge-based system
KPI	Key Performance Indicator
MBSE	Model-based systems engineering
MOKA	Methodology for Knowledge-based Engineering Applications
PAPS	Product Architecture Parameter Synthesis
PAR	SysML Parametric Diagram
PDM	Product Data Management
pdVSM	Product development value stream mapping
PKG	SysML Package Diagram
PLM	Product Lifecycle Management
REQ	SysML Requirements Diagram
ROI	Return on Investment
SysML	Systems Modeling Language

1 Introduction

Despite its potential value with respect to design performance improvement, design automation is applied too little in industry and its benefits are seldom validated in industry (Cagan et al. 2005; Stjepandić et al. 2015). Design Automation (DA) refers to the application of computational methods and tools to support various aspects of the design process by automating design tasks to save time, generate alternative design solutions, explore solution spaces and reuse engineering knowledge. Two major fields of research can be identified contributing to design automation, namely Knowledge-Based Engineering (KBE) and Computational Design Synthesis (CDS) that both investigate computational approaches to support design tasks by means of automation. Design automation tasks refer to the automation of design tasks by application of design automation methods. In this respect, KBE was founded in the 1980s as a merging technology of knowledge based systems (KBS) and computer-aided design (CAD) (La Rocca 2012). Yet, recent definitions account for KBE as any type of automation of repetitive tasks in product development based on the reuse of knowledge (Verhagen et al. 2012). In this work, the focus is on KBE methods relying on Computer Aided technologies (CAx) that are occasionally applied in industry to support configuration in sales and engineering (Zhang 2014; Willner, Gosling, and Schönsleben 2016). On the other hand, CDS methods considered within this thesis, focus on design automation tasks that aim to support the early design stages by generation of design alternatives and solution space exploration based on computationally encoded knowledge (Chakrabarti et al. 2011).

A discrepancy between state-of-the-art in research and state-of-practice can be recognized. This defines the motivation for this research. Reasons for the scarce use of design automation in practice can be broken down into the following aspects:

First, the design practitioners' **lack of knowledge of the opportunities** of state-of-the-art design automation methods offer (Rigger and Vosgien 2018). This is amplified by the lack of interaction and exchange of practices in the research field of design automation. KBE and CDS have the same scientific foundation and shared overall goal of supporting design practitioners by incorporation of knowledge and information in systems and applications that support design. Still, review articles from the fields KBE and CDS often show distinct vocabularies and only few interrelations (Verhagen et al. 2012; Stjepandić et al. 2015; La Rocca 2012; Chakrabarti et al. 2011; Cagan et al. 2005). As a result, there is a need for consolidation of the research field to provide a comprehensive overview of design automation opportunities to support identification of design automation use cases in design practice.

Second, there is a **lack of integration of design automation and their technological environments within the design process** (Anderson et al. 2018; Stjepandić et al. 2015;

Chakrabarti et al. 2011). The application of design automation is intended to support distinct design tasks rather than complete automation of the design process (Dym and Brown 2012). Hence, to succeed in this, the scope of automation needs to be assessed and carefully evaluated. Further, design automation relies on the reuse of formalized knowledge and generation of designs. Thus, design automation applications interface with third party software as applied in design practice such as CAx and information systems. Consequently, design automation integration into design practice needs to take into account the design process comprehensively with the design activities, its supporting methods as well as the related tools and technologies. In this respect, Product Lifecycle Management (PLM) accounts for an integrated approach for the management of product data, engineering workflows and supporting tools and technologies over the whole of a product's life cycle (Abramovici 2007). Thereby, PLM also encompasses the aspects of alignment of the design process and its technological environments. As a consequence, a lack of integration of design automation with PLM results in duplication of data, outdated data and eventually rework (Stjepandić et al. 2015; Cho et al. 2016; Ćatić and Malmqvist 2007).

Third, there is a substantial **shortcoming of methods for** systematic **estimation of the impact of design automation on design practice** (Verhagen et al. 2015). Current practice relies on communication of the value of design automation based on high-level design automation drivers. These account for the motivational aspects for design automation implementation in practice. Design automation drivers include design efficiency, design effectiveness and intangible aspects such as knowledge management (Amen, Rask, and Sunnersjö 1999; Cederfeldt, Elgh, and others 2005; Rigger and Vosgien 2018). Occasionally, quantification of the value of design automation application regarding time savings for repetitive tasks in product development is reported. Yet, this does not account for the full implications of design automation in practice and also other dimensions of design performance such as implications on knowledge management need to be considered. Therefore, a lack of systematic **methods for impact estimation** of **design automation** as well as the need for a **comprehensive metrics system** is recognized.

Finally, **considerable efforts are required for knowledge acquisition** and design automation appears as a **black box** to designers since knowledge is hard-coded in design automation applications. The first refers to high investment costs that makes it difficult to justify the implementation of design automation in industry (Stokes and MOKA Consortium 2001). The common practice of using knowledge engineers for acquisition of knowledge from designers for design automation task formalization (Verein Deutscher Ingenieure 2017) is expensive and often considered critical by designers since it relies on knowledge sharing and trust (Asrar-ul-Haq and Anwar 2016). Closely related, the black box perception of design automation solutions originates from hard-coded knowledge in the design automation application (Plietz 2010) that cannot be validated by the designers since it is not visible to them (P. Bermell-Garcia and Fan 2008). Hence, there is a lack of methods that focus on enabling designers to formalize design automation task themselves to reduce the efforts for design automation task formalization and enable validation of formalized knowledge.

1.1 Objectives and Contributions

The **aim** of this thesis is to **methodologically support design automation task definition** to increase the rate of design automation application in industry. The methodological support focuses on the efforts prior to design automation implementation with particular focus on the needs of design practitioners. This leads to the following objectives: first, a consolidation of KBE and CDS research is required to provide a comprehensive and homogeneous overview of design automation use cases to practitioners. To make the prerequisites for design automation understandable to designers, focus will be put on communicating the use cases based on indication of the product knowledge required for design automation task formalization. Second, the identification of design automation use cases in design practice needs to be supported based on design process decomposition and analysis of product knowledge. The designers' viewpoints on design need to be taken into account as well as the tools and technology perspective to support the integration in design practice. Third, appropriate measures for impact estimation of design automation need to be identified so as to comprehensively assess implications of design automation implementation on design practice. Fourth, means to measure and quantify the impact of design automation prior to implementation are to be defined and, finally, a method enabling designers to formalize design automation tasks themselves needs to be developed.

Due to the collaborative character of design, the methodological support needs to feature a collaborative character to take into account design from various viewpoints. In contrary to existing methodologies for implementation of design automation such as MOKA (Stokes and MOKA Consortium 2001) or VDI 5610 (Verein Deutscher Ingenieure 2017), the focus in this work is put on the viewpoint of design practitioners. This enables implementation of design automation solutions that fit the designers' needs and best support design practice.

1.2 Expected Contributions

The following **three main contributions are expected** in the frame of this work:

The **first expected contribution** relates to the **consolidation of research related to design automation task definition**. A comprehensive literature review on design automation methods from the fields KBE and CDS serves as a basis for identification of means to align research in the two fields from a technological perspective. Design automation task characteristics are identified and an overview of design automation opportunities is provided. Further, literature from different fields related to design process analysis, design performance assessment and knowledge formalization is reviewed and consolidated to establish a meta-model that lays out the vocabulary for reasoning within the scope of design automation task definition.

Building on this, the **second expected contribution** is a **methodology for design automation task definition**. Particular focus is put on the involvement of designers to account for different perspectives of design and related needs to best integrate design automation in design practice and thereby increase awareness and acceptance of design automation in industry. Further, the methodology features evaluation of possible design automation use cases based on metrics and supports designers in formalization of the design automation task. These aspects address the mitigation of black-box perception of design automation in practice and enabling validation, communication, maintenance and eventually formalization of knowledge for design automation by designers.

The **third expected contribution** addresses the results yielded from application of the methodology to different industrial cases. The application of the methodology for use cases related to the early stages of the design process will provide the insight necessary to **verify potential for design automation application for the early stages of design**. Further, the results will highlight potentials for future research on design automation task definition.

1.3 Research Questions

The impact of this research is derived using the research impact model (Blessing and Chakrabarti 2009) depicted in Figure 1-1. The model contextualizes the desired methodological support, key factors addressed by the support (yellow), measurable criteria for evaluation of the support (white) and the success criterion which reflect the aim of this thesis (green) based on findings in literature as well as assumptions based on experience. The directed edges between the factors indicate how they influence each other. “+” and “-“ show the expected change of magnitude of a factor due to application of the proposed methodology for design automation task definition.

The impact model shows that substantial knowledge with respect to the design process and supporting tools and technologies is fundamental to design automation implementation. The knowledge on the design process fosters identification of design automation use cases and potential issues in design. Further, the analysis of the design process provides the necessary context for derivation of metrics for estimation of the impact of design automation on design practice. Building on the identified use cases and metrics, the design automation potential that refers to the potential impact of design automation on design practice can be estimated. This not only supports the justification of investments for design automation implementation but also supports communication of the value to practitioners to increase acceptance. Similarly, automated interfaces to established design support tools and the corresponding integration to designers' workflows leads to an increase of acceptance of design automation in practice. On the other hand, the intuitiveness of the design automation task formalization needs to be considered so that designers are enabled to formalize the design automation task themselves. Closely linked, the design automation task formalization needs to be validated and understood by designers so as to increase the trust of designers in the design automation tool and mitigate black-box perception. This can be addressed by reuse of already formalized and validated knowledge. Further, reuse of knowledge also decreases efforts for design automation task formalization. This itself contributes to an increased rate of design automation application in practice since efforts for design automation task definition substantially contribute to the initial investment and maintenance cost of design automation tools (van Dijk, Reinier E. C., Wang, Haiqiang, and van Dalen, Frank 2012). Thus, multiple factors to address the overall goal of an increased rate of application of design automation in industry can be addressed. Eventually, design performance can be increased with respect to efficiency, effectiveness as well as related to knowledge management.

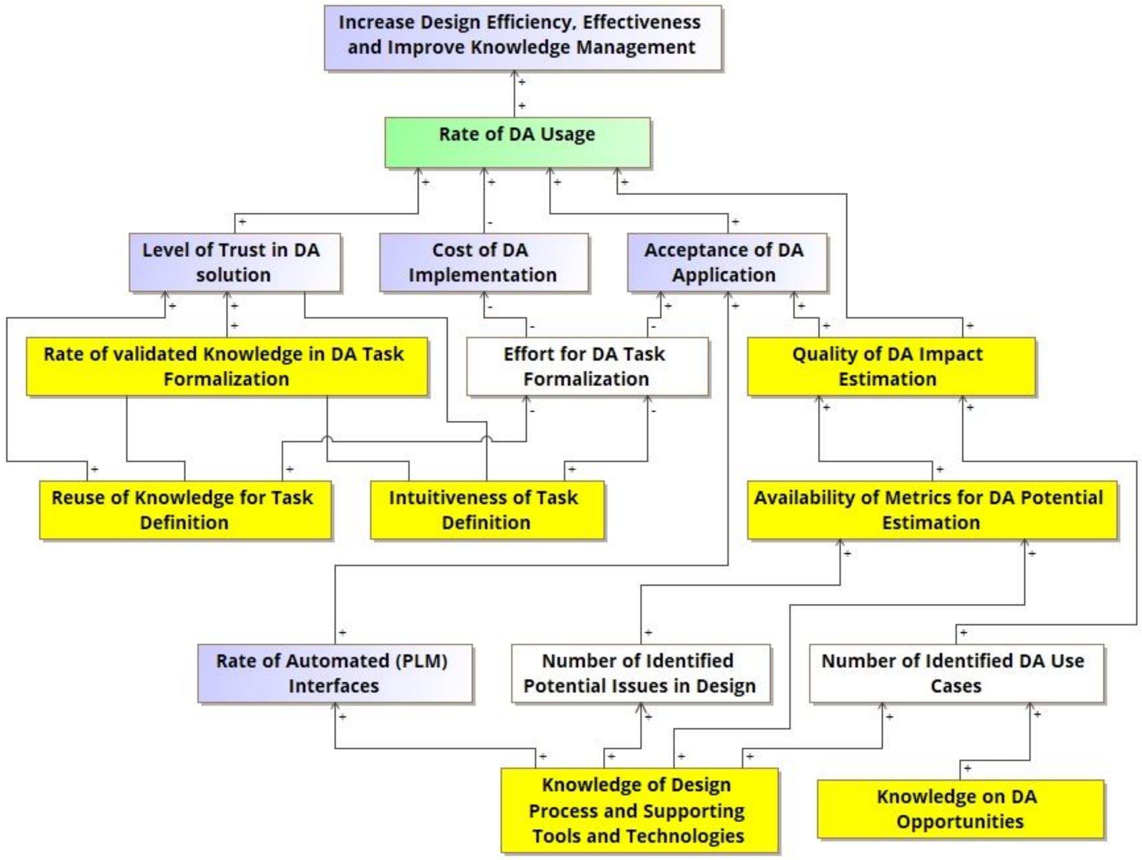


Figure 1-1: Research impact model contextualizing key factors addressed by the developed support (yellow), measurable criteria (white) and the success criterion (green).

In relation to this, the **following main research question** can be posed to validate the research impact model:

> What are the key factors to increase design automation application in industry?

In accordance with the objectives considered in this thesis, the main research questions can be broken down to the following seven **refined research questions**:

1. What are the characteristics of design automation tasks that enable identification of design automation opportunities in design practice?
2. What knowledge is necessary for the identification of design automation use cases in design practice?
3. How can we best integrate design automation in industrial practice?
4. What are appropriate metrics to assess the impact of design automation on design performance?
5. How can the impact of design automation on design practice be estimated prior to the implementation?
6. In what aspects does the usage of a graphical modeling language support designers to formalize design automation task themselves?
7. How can the completeness of a task be assessed in order to support designers when formalizing a design automation task?

1.4 Research Methodology

The overall research methodology follows the descriptive-prescriptive-descriptive approach as proposed in the Design Research Methodology (DRM) (Blessing and Chakrabarti 2009) and is depicted in Figure 1-2. After the clarification of the research context as introduced above, the Descriptive Study I is conducted to further elaborate on the topics and resulted in the design automation task categorization that provides an overview of design automation use cases based on design task characteristics. Further, the design automation metrics library is yielded that summarizes metrics from literature that can be used for assessment of the impact of design automation on design practice. Building on these results, the methodology for design automation task definition is defined within the Prescriptive Study I. It can be seen that it is composed of four supporting methods addressing different aspects of design automation task definition. Finally, the evaluation of the proposed methodology is conducted within the Descriptive Study II. In particular, three different industrial use cases and related user studies are used for evaluation of the proposed methodology and its supporting methods.

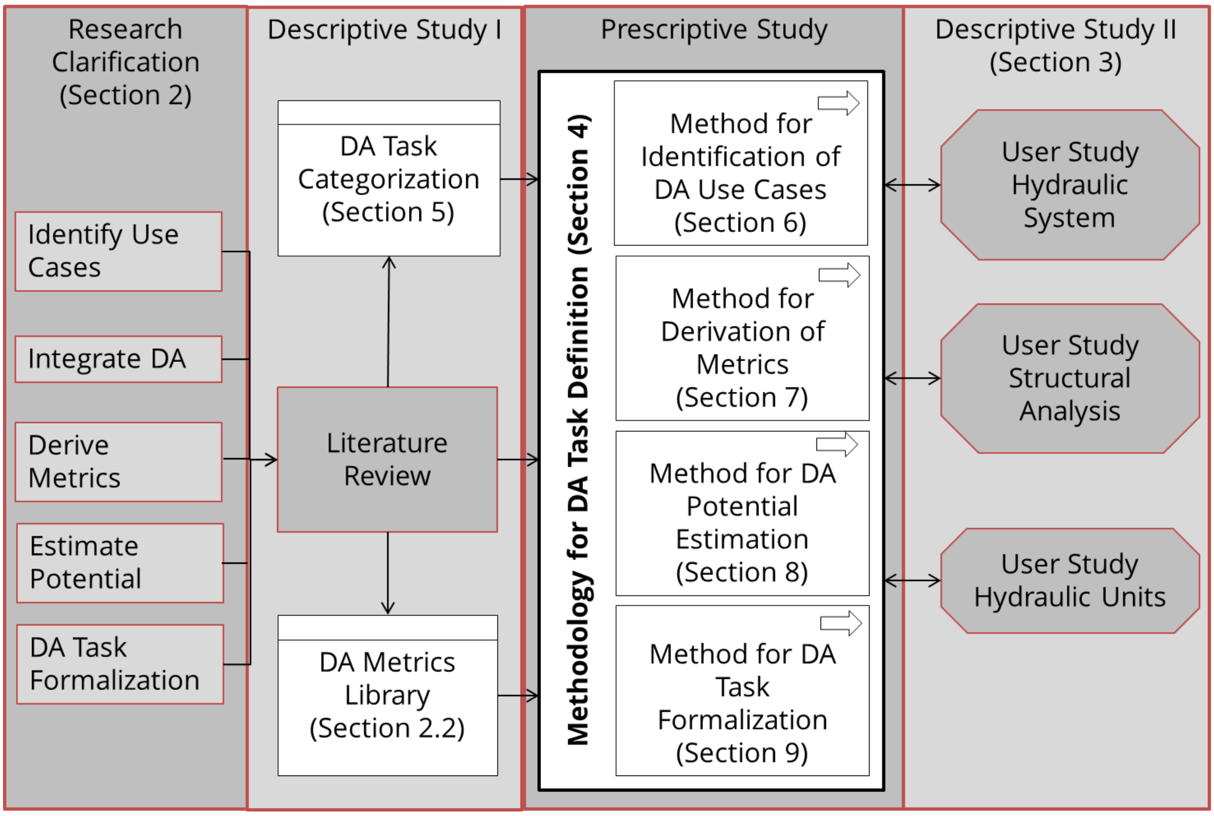


Figure 1-2: Overview of the research methodology as pursued in this thesis and indication of related Sections of the thesis.

1.5 Structure of the Thesis

Section 2 introduces and reviews the related work to provide the background for the research presented in this thesis and elaborate on research gaps. First, design automation is positioned in the context of the design processes and tasks. Next, existing work on identification of design automation use cases and integration of design automation to design practice is analyzed. After this, methods for impact estimation of design automation are reviewed including methods for derivation of metrics. Also, existing sets of metrics for design performance assessment are investigated. Following this, knowledge formalization techniques as well as existing standards for knowledge formalization are reviewed. Finally, a summary section concludes the background section and presents the research gaps addressed in this thesis.

Section 3 presents the industrial cases that are used in this work for evaluation of the proposed methodology and supporting methods. Following this, Section 4 presents the methodology for design automation task definition as developed in this thesis. The four supporting methods and the design automation task categorization that constitute the main building blocks are introduced and how each of them is interrelated to achieve the desired design support. To enable reasoning in the context of the provided methodology, a meta-model relating the main concepts and vocabulary used in this thesis is introduced. Finally, the industrial cases for step-by-step evaluation of the support are presented.

Section 5 presents the derivation of the design automation task categorization as well as the yielded results that build the basis for the developed methodology. It is shown how the design automation task categorization can be used as a basis for derivation of templates with respect to the design process, motivational aspects for design automation application as well as knowledge formalization. After discussing the results, the section closes with providing an answer to research question one. Following this, Sections six to nine introduce the supporting methods that coin the methodology for design automation task definition. It has to be noted that each method is presented as a stand-alone method so they are also applicable when not used in the context of the methodology for design automation task definition. For each method, initially the general context is provided and the main steps are introduced as well as the systematics for its application, e.g. workshop setups. Regarding industrial evaluation, for each method, the industrial settings of the experiments as well as the corresponding results are presented. The results are then discussed in the context of design practice and research as well as with respect to the research questions.

Section 10 discusses the pursued research methodology and the implication of the conducted research on design practice and research. Further, the overall research question and contributions of the conducted work are highlighted before limitations and future work are presented. Finally, Section 11 concludes this thesis by a summary of the findings and puts them into context with the aims and objectives of this thesis.

2 Related Work

This section is a review of related work in the context of design automation task definition. First, Section 2.1 reviews design automation as an integral part of a design process and design activity: The terminology of design automation, design automation methods, design automation tasks and its relation to design activities is clarified. Further, existing concepts for identification of design automation use cases based on categorizations of design automation tasks and decomposition of design processes are analyzed. Also, state-of-the-art on integration of design automation to design practice is investigated to clarify requirements for successful design automation integration with designers' workflows as well as the related tools and technologies. Section 2.2 reviews the related work on assessment of design automation in industrial practice: existing methods for design automation impact analysis and related work on measurement of design performance are critically investigated. After, methods for derivation of a metrics are analyzed and literature on metrics for design performance and related fields is investigated. This builds the basis to establish a metrics system for impact estimation of design automation. To indicate existing shortcomings and provide the necessary context for derivation of a method for design automation task formalization conducted by designers, related work on knowledge formalization techniques and modeling languages is reviewed in Section 2.3. Finally, Section 2.4 summarizes the background and highlights the research gaps identified addressed in this thesis.

2.1 Design Automation

This section first clarifies the terminology of design automation in the context of the design process. Second, approaches for identification of design automation use cases based on design task categorizations are reviewed. Finally, aspects of design automation integration to design practice are analyzed including systematics for integrated design process modeling taking into account the tools and technology perspective of design support methods.

2.1.1 Design Automation Terminology

Research on *design automation* has proliferated in the two major research fields KBE and CDS that both investigate computational methods to support design tasks, further referred to as *design automation methods*. These methods are developed to address specific types of design tasks and are characterized by the applied knowledge formalization and reasoning techniques. *Design automation tasks* correspond to design tasks that are to be automated based on the implementation of a design automation method, i.e. a *design automation application*. It has to be noted, that other viewpoints exist that consider design automation as a branch of KBE that focuses on macro-based parameterization of CAD models (van der Velden, Bil, and Xu 2012; Verein Deutscher Ingenieure 2017). As introduced before, in this work both KBE and CDS are considered contributing to the overall research field design automation.

In the context of design problem solving, a *design problem* can be considered as being composed of multiple *design tasks* (Chandrasekaran 1990). Similarly, the *design process* that supports solving a design problem can be decomposed. In fact, the design activities associated to a design process can be broken down over several levels to “elementary operations” such as reading, sketching, speaking, dimensioning (Hubka and Eder, 1982). In an attempt to further analyze design activities, Sim and Duffy (2003) reviewed literature on design methods and reported activities in order to identify a set of generic design activities. The analysis of the knowledge levels for conducting a design activity allowed them to attain a set of generic design activities with respect to design states. In this context, the principle of knowledge levels permits to distinguish the knowledge required to conduct an activity and the related processing of knowledge (Newell 1981). Based thereon, Sim and Duffy (2003) describe *design activities* based on *input knowledge, output knowledge, the corresponding goals* as well as the related *resources*. The desired output is specified to reflect the goals to a certain degree of effectiveness (Duffy 2005). The goals are deduced from the initial requirements and constraints or derived from changes that occur within the design process. Thus, goals guide and direct the design activity. Putting into context a design activity and a design task, a design activity is carried out to meet the desired output specified within a design task (Duffy 2005). If mapping these definitions to the context of design automation, a design automation activity addresses a design automation task and the required resources refer to the design automation application as well as the designer using the respective application. However, in contrary to a design task, the design automation task requires formalization of goals to explicitly capture the knowledge needed for guiding reasoning when applying a specific design automation method. Hence, ill-defined design problems with vague or incomplete goals (Dinar, Danielescu, et al. 2015) cannot be considered design automation tasks. Finally, the output of a design automation task needs to be specified to indicate the change of design states. This impacts the type of input knowledge that is necessary to enable the desired transformation.

Summarizing, a general *design problem* is methodologically supported by a *design process* that can be decomposed in one or many *design tasks* and *design automation tasks*, respectively. These are themselves associated to the design process. It can be seen, that the design automation task distinguishes itself from a general *design task* by the prerequisite for formulation of goals as well as the predefined set of resources required for automation.

2.1.2 Identification Design Automation Use Cases

Existing categorizations of design tasks for identification of design automation use cases based on analogy are either ambiguous or too high level. For example, Brown and Chandrasekaran (1990) characterize design tasks by means of availability of domain knowledge and problem solving strategy and introduce three different categories that are then used for determining available design automation methods: routine, innovative and creative design tasks. Tong and Sriram (1992) further refine this categorization and map design automation methods to routine and innovative design tasks. Similarly, Chandrasekaran (1990) analyze design tasks and define three generic classes, i.e. “propose, critique and verify”. Due to the generality of this categorization, each design task can be understood as an instance of any of these classes or of combinations of them. Regarding the approaches presented above (Brown and Chandrasekaran 1990; Tong and Sriram 1992; Chandrasekaran 1990), the generality of classes and their instances that remain confined to higher levels of abstraction considering design task modeling do not allow to derive any specific guidance for identification of design automation opportunities in practice.

With the intention to systematically analyze and categorize computational methods for solving design tasks, quantitative and qualitative criteria as well as different categories of design tasks were introduced by Cagan, Grossmann, and Hooker (1997). However, they state that the introduced categorization is “rather general” and is intended as a framework for the representation of tasks rather than serving as a practical means to identify similar design tasks to a given one in the context of design automation. Building further, Patel and Campbell (2010) categorize embodiment design tasks based on the availability of evaluation techniques during the solution space search process and the relation of parameters to changes within the topology. Thus, appropriate search/optimization algorithms to tackle the parametric topology change can be determined for this specific type of task. However, their categorization solely focuses on one specific type of design automation task and needs to be extended with respect to additional design automation tasks addressing other aspects of the design process as well as the corresponding methods. With a focus on computational support for formalizing optimization problems, Witherell, Krishnamurty, and Grosse (2007) introduced an ontology-based tool for supporting the definition and implementation of optimization problems by means of a taxonomy of optimization methods. Though, the approach is limited to optimization tasks and as such omits key characteristics of optimization tasks which are relevant for proper assignment of an adequate optimization method. Based on the design task characteristics introduced by Cagan, Grossmann, and Hooker (1997), Amen, Rask, and Sunnersjö (1999) analyze twelve industrial case studies of design automation implementation for derivation of a mapping of design task characteristics to design automation methods. Yet, the results solely provide high-level information on automation possibilities as the categorization only maps to programming paradigms (procedural vs. declarative), optimization and case-based reasoning techniques. A more systematic investigation of computational tools that support the conceptual design is conducted in (Jensen 1999). However, the investigations there address the conceptual design stage, only. In an effort to achieve more generality considering the design process context, Rigger, Münzer, and Shea

(2016) present design automation task categories according to their occurrence within the design process and spatial attitudes, i.e. whether embodiment is considered or not. However, only contributions from the research field CDS were considered and a detailed formal description of design automation task categories was not given.

Summarizing, the relevant literature on identification of design automation use cases shows that a comprehensive categorization that explicitly considers design automation tasks in the context of the design process is needed. Such a categorization enables mapping of design automation methods to design processes based on the design automation task characteristics. Therefore, relevant product knowledge as well as corresponding design states need to be identified and characterized for design automation task analysis. Existing design task categorizations need to be expanded to account for particularities of design automation tasks. Therefore, the level of detail needs to be increased by explicit information that is necessary for the mapping to state-of-the-art design automation methods and corresponding knowledge formalizations.

2.1.3 Design Automation Integration

Best practice methods for implementation of knowledge-based engineering type of design automation rely on contextualizing the intended support with the design process, e.g. (Verein Deutscher Ingenieure 2017). Thereby, focus is on identification of the knowledge intensive tasks and availability of knowledge carriers, e.g. related to domain experts or explicit formalizations that are available. More generally and with a focus on reusability of design automation task related knowledge, Yu, Cha, and Lu (2012) and J. H. Panchal et al. (2009) propose a hierarchical decomposition of a design process to the level of granularity that enables re-use of knowledge based on declarative design automation task templates with well-defined inputs and outputs. With the intention to related KBE or engineering knowledge in general to the design process, Pablo Bermell-Garcia et al. (2012) define enterprise knowledge resources that contain information with respect to product knowledge, process knowledge and case specific data. Product knowledge accounts for any type of knowledge related to a product and can also encompass KBE applications. The process knowledge can be applied for specification of (automated) workflows in design. The application of both process and product knowledge to specific cases is then stored in the case entity to support a learning process among users. However, despite putting into context the design process and design automation, none of these approaches address integration from a tools and technology perspective. In fact, even though efforts addressing more systematic integration exist (Fan and Bermell-Garcia 2008), only few works report the systematic integration of design automation and PLM enabling technologies (Di Gironimo 2014; Gerhard and Lutz 2011; Lund, Fife, and Jensen 2005).

In their work, Whitfield et al. (2011) presented a top-down and bottom-up approach for development of a common data structure to exchange information among geographically distributed design departments. Whereas top-down is applied to account for the design process perspective, the bottom-up approach considers the interface specifications of design (automation) tools. The systematic is shown to be appropriate for identification of interfaces and definition of integration specifications. Yet, no method for documentation of the gathered information is presented that permits the simultaneous consideration of design processes and corresponding tools and technologies in an integrated manner. With this respect, Enterprise Architecture (EA) provides principles, methods and models for design and re-design of a “business enterprise” (Bernus, Nemes, and Schmidt 2003), or, in more detail, an “enterprise’s organizational structure, business processes, information systems and infrastructure” (Lankhorst 2017). More than 30 EA supporting methods and standards have emerged since the 1980s (Bernus, Nemes, and Schmidt 2003). Most notable is the Zachman Framework (Zachman 1987) that accounts for various viewpoints when developing architectures. Other representative examples include the TOGAF (The Open Group 2009) that enables systematic development of architectures and ISO/IEC/IEEE-42010 (ISO/IEC/IEEE 2011, 42010), which provides vocabulary and methods for description of systems and software architectures. Referring to design as an integral part of an engineering enterprise, the systematics of enterprise architecture are also applicable in a design context, in particular when considering PLM (McKendry, Whitfield, and Duffy 2015). Hence, in this work methods of enterprise architecture are used for the systematic alignment of design activities, applications of design process support methods and corresponding tools and technologies within the design process.

The ArchiMate graphical modeling language (The Open Group 2016) has been developed as a standardized language for enterprise architecture modeling and communication. It has been proven to be useful for specifying integration and interoperability (Chen, Doumeingts, and Vernadat 2008) and features modeling in three layers: business, application, and technology. As described above, these layers can be mapped to the design process with respect to design activities, applications of design process support methods and corresponding tools and technologies. ArchiMate features task precedence models that are commonly used as analytical modeling approaches for communication, participation and reengineering of business processes (Melao and Pidd 2000). The applied flowchart-style notation of task precedence models have been shown to be easily readable and thereby support communication of the content with stakeholders as well as permit the necessary flexibility with respect to modeling the appropriate level of granularity of a design process (Wynn and Clarkson 2018). In order to account for the situated (Gero and Kannengiesser 2004) and iterative (Wynn and Eckert 2016) character of design processes, a higher level of uncertainty is included in the design process models as well as tasks that possibly trigger iterations. In contrary to other standardized notations for the task precedence modeling, e.g. BPMN

(White and Miers 2008) or IDEF3 (Mayer et al. 1995), ArchiMate permits to systematically take into account the tools and technology perspective.

Summarizing the efforts of design automation integration to design practice, it can be concluded that both the design activities and related technological environments need to be taken into account for successful integration of design automation within design processes. ArchiMate as a standardized language enables to capture these aspects of design automation implementation in an integrated manner. However, systematics for identification of knowledge and interfaces for design automation implementation still need to be developed which is subject of this work.

2.2 Estimation of the Impact of Design Automation on Design Practice

This section critically reviews related work on impact estimation of design automation. First, methods aiming at impact estimation of design automation are analyzed before the background for metrics derivation in the context of design automation is presented. In this respect, Section 2.2.2 examines the basic characteristics of measurable criteria needed for assessment of the impact of design automation and clarifies the vocabulary used in this work related to design performance. Next, Section 2.2.3 reviews methods pursuing a top-down strategy for metrics definition from both the software development and engineering design domain to derive requirements for metrics derivation in the context of impact estimation of design automation. Finally, Section 2.2.4 evaluates existing sets of drivers for design automation implementation, as well as literature on established sets of metrics in the context of design performance and software quality assessment. This allows relating the drivers as motivational aspects for design automation implementation with existing sets of metrics.

2.2.1 Methods for Impact Estimation of Design Automation

There is a substantial lack of methods for systematic impact estimation of design automation that enables comprehensive assessment of the value of design automation so to justify efforts for design automation implementation in practice (Verhagen et al. 2015). Existing efforts aim at justification of design automation implementation based on the routine character of a design task (Emberey et al. 2007), time savings and return on investment (ROI) estimation based on complexity assessment of a potential design automation solutions (van der Velden, Bil, and Xu 2012). However, these approaches do not provide the necessary details of estimation of implementation efforts. Addressing this shortcoming, (Mulder et al. 2015) present an approach that is based on the type of activity the automation is implemented for and the existing and desired levels of automation. A classification of activities as well as a listing of different levels of automation that can be attained for each activity is proposed. Based thereon, a cost coefficient value for each activity and level of automation is determined based on experience. In combination with discrete event simulation of the design process and multi-objective optimization to investigate different levels of automation, a Pareto front with respect to lead time and investment cost is determined for various automation scenarios. In a similar manner, Pal and Ghosh (2017) assess efforts for software implementation in design and manufacturing processes based on the use case points method and related complexity assessment. Complexity of processes is assessed with respect to actors/interfaces, e.g. human-computer, system-system, etc., and the number of transaction for each use case, such as events between an actor and a systems or design activities. These measures in conjunction with weighting factors permit to estimate the effort for implementation of software tools based on the hours needed for implementation per use-case point. Despite being more systematic, the approaches by Pal and Ghosh (2017) and (Mulder et al. 2015) strongly rely on empirical historical data for estimation of durations of design activities and implementation efforts. This makes it difficult to make reliable effort estimations due to the heterogeneity of design automation tasks. Further, in their work, focus is put on the isolated assessment of the design automation task and related time savings for ROI analysis, rather than its integration to the design process and respective impacts on design performance in general.

Taking into account the design process perspective, Schut, Kosman, and Curran (2013) define high-levels objectives regarding cost, time and quality and related KPIs based on interviews with engineers. Following this, KPI measurements and process analysis with respect to value stream analysis are conducted to prioritize and refine the objectives. This prioritized list is then compared with initiatives addressing these objectives and corresponding effort estimations to enable informed decisions with respect to design automation implementation. Whereas the approach presents a systematic for justification of design automation implementation, it lacks details on identification of design automation use cases and definition of appropriate metrics for estimation of the impact. Building upon this approach, Verhagen et al. (2015) propose to analyze engineering processes with respect to information flows. Thereby, (design) automation potential is quantified based on assessment of information waste in product development. However, information waste is difficult to account for as the sole measure of efficiency in design, due to the iterative and situated character of design processes (Wynn and Clarkson 2018).

Hence, more comprehensive assessments are to be considered for impact estimation in design, accounting for both the design process perspective as well as the respective tasks. Thus, impact estimation of design automation needs to be done on a case-by-case basis and requires identification of design automation use cases; selection of appropriate metrics; and, making informed decision about the impact of design automation implementation on selected metrics so to quantify the impact of design automation implementation on design performance.

2.2.2 Characteristics of Design Performance Metrics

Metrics are means to assess an issue in either quantitative or qualitative form (Horváth, Gleich, and Seiter 2015). With respect to engineering design, the work by Robinsons (2016) urges the need for reliable and valid metrics for quantitative assessment of design. Thereby, reliability refers to consistency of results for repeated measurements and validity corresponds to the actual measurement, whether a metric assesses what it is supposed to measure and not any possible influencing factors. Within the work of Duffy (2005), metrics for design performance assessment are defined to be Numeric, Explicit, Appropriate, and True (NEAT). In the context of reliability and validity, the characteristics "true" and "appropriate" can be used as synonyms, respectively. Considering further characteristics of metrics, metrics can be *accumulative*, i.e. the sum of sub-metrics, *derived*, i.e. calculated based on sub-metrics and a particular equation or *independent*, meaning, directly measured (Duffy 2005). With respect to scaling, metrics can be based on an absolute scale or comparative scale for the case that multiple measurements and no reference scale are available (Kreimeyer and Lindemann 2011).

Regarding the actual documentation of metrics, the format used in ISO/IEC 25022-25024 (ISO/IEC 2016a, 2016b, 25023, 2015, 25024) standards will be applied for comprehensive documentation. Hence, information needs to be provided for each metric with respect to:

- ◊ A category for identification of appropriate metrics,
- ◊ Descriptions what the metric is for and how to calculate the metric.

These guidelines for metrics documentation are equally applicable in the context of design performance metrics.

2.2.3 Top-down Derivation of Metrics

The Goal-Question-Metric (GQM) method has been coined in the 1980s, in particular for the domain of software development (Basili 1992). The first step of the method comprises the definition of project goals that define the purpose of a project, what aspects are considered and in which environment/ context the project takes place. Questions are then generated to define the goals in a quantifiable way so that metrics can be defined that enable the actual measurement of goals achievement. In the context of engineering design, Duffy (2005) further elaborates on NEAT metrics with respect to a top-down methodology for derivation of metrics that is composed of, first, design activity goals identification, definition of metrics for assessment of the activity goal achievement, definition of data as well as measurement methods and, lastly, specification of targets that specify desired performance. More particularly, for research on design automation method development and integration, Jensen (1999) proposed a mixed theoretical and empirical approach with iterative empirical, qualitative assessment of the proposed design automation method. In contrast to this, Bracewell et al. (2001) propose a top-down quantitative approach for systematic development and assessment of design automation methods. The approach is based on the Design Research Methodology (Blessing and Chakrabarti 2009) to determine success factors (goals), impact chains (questions) and measurable criteria (metrics) so to assess whether a newly developed design automation method meets the goals in an industrial context. The approach by Bracewell et al. (2001) is demonstrated based on a proof-of-concept implementation of a structural shape annealing method. Yet, specific systematic for determination and identification of appropriate metrics is omitted.

From a lean engineering perspective, McManus (2005) introduce the concept of product development value stream mapping (pdVSM) for identification of bottlenecks in design processes as well as for the derivation of metrics for quantification of value and waste in design. Considering design as knowledge intensive process, non-value added tasks and waste are related to information flows along 7 dimensions: waiting, inventory, over-processing, over-production, transportation, unnecessary movement and defective products. The value of the investigated (part of) design process needs to be determined individually based on the goal the process aims at delivering. Thus, potential for process improvement can be identified and individual design tasks quantitatively assessed. More qualitatively, Chao and Ishii (2007) propose to apply the failure mode and effects analysis (FMEA) for systematic analysis of possible failure modes / design errors and corresponding causes and effects in design, i.e. design process FMEA (dpFMEA). Failure modes are classified along the dimensions of knowledge, analysis, communication, execution, change and organization. This classification is used as a basis for derivation of questions that are posed in workshops for identification of potential failure modes and related causes. The importance of failure modes that possibly occur in the design process, likelihood of occurrence of failure modes as well as likelihood of detectability of the failure modes in design are qualitatively rated so to calculate the error priority number, which ranks the identified failure modes and causes according to the risk to harm design. Focusing on systematic integration of tools and technologies in design, Morgan and Liker (2006) propose to view lean product development as an integrated system of people, processes and tools. Still, most of the studies addressing lean product development do not equally consider these aspects but are either people-oriented (Garcia and Drogosz 2007), process efficiency-oriented (Gautam and Singh 2008; Wang et al. 2011; Tyagi et al. 2015), or tools and technology oriented (Thomke and Fujimoto 2000; Bogusch et al. 2016). Tribelsky and Sacks (2011) and Verhagen et al. (2015) focus on identification of waste in information flows and processing, where the latter assesses waste so to quantify and justify the implementation of automation for "engineering lifecycle tasks” as a means to reduce waste. In a method combining lean engineering and 6σ Sigma (Vosgien et al. 2011; Vosgien 2015), it is proposed to apply the Define Measure Analyse Design Verify methodology (DMADV) for the integration of tools and technology in product development through systematic consideration of processes, tools as well as the corresponding alignment of processes and tools. In particular, value stream analysis is used for identification of waste sources from a design as well as a tools and technology perspective. Next, metrics are derived based on product development waste drivers as identified in (Bauch 2004). However, no metrics system is presented due to lack of validation of the approach.

Hence, it can be concluded that a validated method for metrics derivation for impact estimation of design automation that considers all aspects of design is missing. Best practice methods from both design and software engineering pursue a top-down approach and start with the definition of goals before the actual definition of metrics for quantification of the goals' achievement (Koziolek 2008). In order to enable a top-down approach for derivation of metrics to assess the impact of design automation on design practice, established methods for identification of issues, causes and related effects in design such as pdVSM and dpFMEA need to be further generalized to cover all aspects of the design process including the tools and technology perspective. Whereas the removal of causes refers to the design automation goals, information on the effects can be used as questions to guide metrics definition for impact estimation of design automation.

2.2.4 Drivers and Metrics

Few works exist that investigated and industrially validated the motivational aspects or drivers for design automation implementation in industry, e.g. (Amen, Rask, and Sunnersjö 1999; Cederfeldt, Elgh, and others 2005; Rigger and Vosgien 2018). Further, the related metrics for quantitative assessment whether design automation provides the expected benefit according to drivers are missing. The only exception is related to time savings achieved by automation. With this respect, multiple studies can be found in literature reporting time saving for repetitive tasks of up to 95% (Bermell-Garcia et al., 2012; Cederfeldt, 2006; Chapman and Pinfold, 2001; Emberey et al., 2007; La Rocca and van Tooren, 2010; Raffaeli et al., 2013; Colombo et al., 2005; Ruschitzka et al., 2010; Danjou et al., 2008; Corallo et al., 2009; Kulon et al., 2006; van der Elst and van Tooren, 2008; van der Laan and van Tooren, 2005).

Hence, in this work a literature review of general design performance measurement and design key performance indicators (KPI) is conducted. In particular, a search based on the initial key words "engineering design" coupled with "performance measurement", "KPI", "metrics", "performance assessment" was performed to search for journal and conference papers (Do 2014; McManus 2005; Gries and Restrepo 2011; Hvam, Anders, and Niels 2010; Costa et al. 2014; Dinar et al. 2016; Cherry and Latulipe 2014; Summers and Shah 2010; Škec, Cash, and Štorga 2017; Kreimeyer and Lindemann 2011). In the context of design automation metrics, the list of drivers introduced in Rigger and Vosgien (2018) are used for organization of the design automation metrics system. In particular, the design automation drivers are organized as a taxonomy according to the dimensions of efficiency and effectiveness (O'Donnell and Duffy 2002), as well as knowledge management to account for non-tangible aspects (Škec, Cash, and Štorga 2017). The latter is relevant for impact estimation of design automation since being a major driver for design automation implementation in industrial practice (Rigger and Vosgien 2018). The attained metrics are mapped to the taxonomy of drivers and are listed in in Table 2-1 with respect to design efficiency, Table 2-2 for design effectiveness and Table 2-3 for metrics related to knowledge management. The tables list both the name of metrics as well as the qualitative description required for calculation, if available. Hence, a preliminary, literature-based set of metrics for design automation assessment is yielded.

In contrary to design performance assessment, norms and standards for software quality assessment exist that can be categorized as metrics for quality in use and product quality (ISO/IEC 25023:2016) (ISO/IEC 2016b, 25023). In particular metrics for assessment of software product operation / quality in use (Fenton and Bieman 2014) are considered relevant for comprehensive assessment of design automation in practice. For instance, a design automation tool permits fast and reliable synthesis of designs; however, the application lacks usability and therefore isn't applied in practice. Thus, the assessment of software quality in use is required for assessment of designer's satisfaction with respect to usability, reliability and efficiency of design automation applications. This is necessary to attain the complementary view of impact estimation of design automation from a technology perspective. Aspects of software quality measurement related to software revision such as code quality metrics (Heitlager, Kuipers, and Visser 2007; Zeiss et al. 2007) are not considered in this work since they have no relation to the actual (engineering) design performance and quality. Even though multiple approaches for metrics definition in the context of software quality exist, e.g. (Seffah et al. 2006), this works uses metrics of the recently released ISO/IEC 25022:2016 (ISO/IEC 2016a, 25022) standards for software quality in use. For the sake of illustration, the categorization used in this standard as well as one representative metric is depicted in Table 2-4. Data quality can be considered an intermediate layer that indirectly impacts both design performance as well as software quality. Metrics from the related standard ISO/IEC 25024:2015 (ISO/IEC 2015) are used in this work and its categorization is summarized in Table 2-5.

Table 2-1: Metrics including a description (if provided by the referenced work) for assessment of design efficiency sorted according to design automation drivers (Rigger and Vosgien 2018). Drivers and sub-drivers are listed in bold letters.

Driver	Metric Name	Metric Description	References
Cost reduction			
	Outsourcing rate	Number of external partners / number of own engineering staff	(Gries and Restrepo, 2011)
	Engineering utilization	Allocated demand of active projects / total available productive capacity	(Gries and Restrepo, 2011;

Driver	Metric Name	Metric Description	References
			Costa et al.,2014)
	Utilization	At the current workload, how much of the capacity is actually needed/used.	(McManus, 2005)
	Development lead time	End date of last development activity minus start date of first development activity	(Gries and Restrepo, 2011; McManus, 2005)
	Time to market of the product		(Costa et al., 2014)
	Time for item development		(Do, 2014)
	On-time delivery	Number of specifications out of the total number of specifications that are completed within the agreed time span	(Hvam, Haug, and Mortensen, 2010; Costa et al. 2014; Gries and Restrepo, 2011)
	Delay time	Number of days exceeding from the expected delivery day	(Costa et al., 2014)
	Time for item approval		(Do, 2014)
	Time for engineering change proposal processing		(Do, 2014)
	Cost per job or recurring costs	What resources are expended to do a job	(McManus, 2005)
	Core process time	Hours (or other time units) of continuous work spent on core task, excluding set up, trouble shooting, information gathering, etc.	(McManus, 2005)
Increase of productivity			
	Engineering productivity	Total work time booked against billable projects / total contractual work time	(Gries and Restrepo, 2011)
	Delay time statistics	Mean and deviation, or distribution, of wait times (best)	(McManus, 2005)
	Employee satisfaction level		(Costa et al., 2014)
	Cycle time	Duration of specific iteration within the process	(McManus, 2005)
	Motivation and satisfaction at individual level	Personal satisfaction and motivation	(Škec, Cash, and Štorga, 2017)
	Motivation and satisfaction at team level	Time spent on non-working social activities, % of activities done with lower motivation and irrelevant activities	(Škec, Cash, and Štorga, 2017)
Improved accuracy of cost calculation			
	Cost performance	Budgeted cost of work performed / actual cost of work performed	(Gries and Restrepo, 2011)
	Schedule performance indicator	Budgeted cost of work performed / budgeted cost of work scheduled	(Gries and Restrepo, 2011)
Improve tendering (Fast reaction to changes of customer requirements / Enable efficient teamwork of design and sales department / Less iterations between sales and engineering department / Faster tendering in order to be the first of competitors to provide an offer)			
	Order delivery lead time	Number of days from when a customer makes an enquiry until the customer receives an offer	(Hvam, Haug, and Mortensen, 2010)
	Resource consumption for making specifications	The frequency of the individual specification activities, combined with the duration (use of man-hours) of the individual specification activities	(Hvam, Haug, and Mortensen, 2010)

Driver	Metric Name	Metric Description	References
	Design process structure: size and density	Number of domains, number of nodes, number of edges, number of classes, number of interfaces between domains, number of edges per node, relational density, number of unconnected nodes	(Kreimeyer and Lindemann, 2011)
	Design process structure: adjacency	Activity / passivity, degree correlation (nodes), degree correlation (edges), degree distribution, fan criticality, synchronization points / distribution points, number of independent sets	(Kreimeyer and Lindemann, 2011)
	Design process structure: adjacency	The number of reachable nodes, reachability of a node	(Kreimeyer and Lindemann, 2011)
	Design process structure: closeness	Proximity, relative centrality (based on betweenness)	(Kreimeyer and Lindemann, 2011)
	Design process structure: connectivity	Node connectivity, edge connectivity	(Kreimeyer and Lindemann, 2011)
	Design process structure: paths	Number of paths, path length, weight of an edge, centrality of path (based on centrality), centrality of path (based on degree) degree of progressive oscillation	(Kreimeyer and Lindemann, 2011)
	Design process structure: hierarchies	Height of hierarchy, width of hierarchy, tree criticality, snowball-factor, forerun-factor, tree-robustness, maximum nesting depth	(Kreimeyer and Lindemann, 2011)
	Design process structure: clustering	Number of cliques, cluster-coefficient (local), cluster-coefficient (global) module quality 1 (flow of information), module quality 2 (compactness)	(Kreimeyer and Lindemann, 2011)
	Design process structure: cycles	Number of cycles, number of cycles per node, number of cycles per edge, number of feedbacks, activation of cycle, number of starting points for iterations, iterative oscillation	(Kreimeyer and Lindemann, 2011)
	Design process structure: several domains	Bipartite density, number of organizational interfaces	(Kreimeyer and Lindemann, 2011)
	Design process structure: cognition	Cognitive weigh, degree of non-planarity	(Kreimeyer and Lindemann, 2011)
	Design process structure: boolean operators	Mccabe cyclomatic number, control-flow complexity	(Kreimeyer and Lindemann, 2011)
	Size complexity	Complexity as function of process, product and design problem	(Summers, and Shah, 2010)
	Number and nature of bottlenecks		(Costa et al., 2014)
	Automate routine tasks (Reduction of time spent for repetitive tasks / Free resources from routine tasks so that time can be spent on creative/innovative tasks / Automated generation of documentation)		
	Innovativeness and ideation capacity at individual level	% of relevant ideation activities regarding product, process and other domains by any individual	(Škec, Cash, and Štorga, 2017)
	Innovativeness and ideation capacity at team level	% of relevant ideation sessions during team activities regarding product, support by the manager for innovation	(Škec, Cash, and Štorga, 2017)
Manufacturing Cost			
	Carry-over components percentage		(Do, 2014)

Driver	Metric Name	Metric Description	References
	Purchased components percentage		(Do, 2014)
	Support of production planning		

Table 2-2: Metrics including a description (if provided by the referenced work) for assessment of design effectiveness sorted according to design automation drivers (Rigger and Vosgien 2018). Drivers and sub-drivers are listed in bold letters.

Driver	Metric Name	Metric Description	References
Error reduction in design			
	Number of approved items		(Do, 2014)
	Number of non-approved items		(Do, 2014)
	Number of engineering changes		(Do, 2014; Costa et al, 2014)
	Number of engineering change request		(Do, 2014)
	Number of approved drawings		(Do, 2014)
	Number of non-approved drawings		(Do, 2014)
	Engineering first pass yield	Number of functional components submitted without rejection / total number of functional components	(Gries and Restrepo, 2011)
Product Quality			
	Solving complex design tasks		
	Rework rate	Incident of defects in the process output	(McManus, 2005)
	Generation of optimized solutions		
	Generation of alternatives		
	Number of solutions developed in parallel	# of evaluated alternative designs per design activity / average # of solution alternatives investigated per activity	(Costa et al., 2014)
	Ideation quantity	Total number of generated ideas	(Dinar, et al., 2016)
	Ideation variety	Total number of unique ideas	(Dinar, et al., 2016)
	Ideation novelty	A measure of how rare the generated idea is	(Dinar, et al., 2016)
	Ideation quality	Measures the feasibility of an idea and whether it meets the design requirements	(Dinar, et al., 2016)
	Ideation quality	Creativity support index	(Cherry and Latulipe 2014)
	Enable development of customer specific solutions		
	Customer satisfaction	Measured many ways, e.g., requirements checklist, survey	(McManus, 2005)
	Customer satisfaction	The level of customer satisfaction with the product given to him	(Costa et al., 2014)
	Enhanced manufacturability		
	Improved consistency of designs (according to design guidelines etc.)		

Table 2-3: Metrics including a description (if provided by the referenced work) related to knowledge management sorted according to design automation drivers (Rigger and Vosgien 2018). Drivers and sub-drivers are listed in bold letters.

Driver	Metric Name	Metric Description	References
Establishment of a knowledge base / Capitalization of knowledge			
	Competences and knowledge at individual level	Problem solving ability, decision making, creativity technical knowledge, responsibility	(Škec, Cash, and Štorga, 2017)
	Competences and knowledge at team level	Number of team members with outstanding , insufficient competencies number of non-human resources which are not used	(Škec, Cash, and Štorga, 2017)
	Competences and knowledge at team level	# of concepts discussed in team / # of concepts discussed on average	(Škec, Cash, and Štorga, 2017)
	KW capitalization	# of stored designs / # of designs created	(Cederfeldt & Elgh, 2005)
Reuse of knowledge / Reuse of existing solutions & designs			
	Usage of standard items	Ratio number of standard parts in product design / total number of parts in product design	(Do, 2014)
	Degree of standardization	The percentage of engineered parts / components within a finished product used in at least one other previously finished product	(Gries and Restrepo, 2011)
Fast visualization of designs			

Table 2-4: Categorization of software quality metrics according to ISO/IEC 25022. For illustration, a metric is indicated to assess effectiveness. The column “Driver” refers to the categories.

Driver	Metric Name	Metric Description
Effectiveness		
	Tasks completed	The proportion of the tasks that are completed correctly without assistance
Efficiency		
Satisfaction		
	General satisfaction	
	Usefulness	
	Trust	
	Pleasure	
	Comfort	
Freedom from risk		
	Economic risk mitigation	
	Health and safety risk mitigation	
	Environmental risk	
Context coverage		
	Context completeness	
	Flexibility	

Table 2-5: Categorization of data quality metrics according to ISO/IEC 25024. For illustration, a metric is indicated to assess data accessibility. The column “Driver” refers to the categories.

Driver	Metric Name	Metric Description
Data Accuracy		
Data Completeness		
Data Consistency		
Data Credibility		
Data Currentness		
Data Accessibility		
	User accessibility	Number of data items relevant to the user's task and accessible / total number of data items relevant to the user's task
Data Compliance		
Data Confidentiality		
Data Efficiency		
Data Precision		
Data Traceability		
Data Understandability		
Data Availability		
Data Portability		
Data Recoverability		

2.3 Design Automation Task Formalization

Building upon the reuse of formalized knowledge, design automation depends on acquisition, formalization and re-use of engineering knowledge. In this section, first the terminology of data, information and knowledge is clarified and ways to differentiate the types of knowledge are reviewed. Next, existing methodologies for knowledge formalization in the context of design automation are reviewed. Following this, existing efforts for application of the SysML modeling language as a standardized language for design automation task formalization are investigated.

2.3.1 Design Automation Task Knowledge

Knowledge can be classified as tacit and formal knowledge (Chandrasegaran et al. 2013). Whereas the first refers to expert knowledge and intuition, formal engineering knowledge corresponds to information embedded in data, such as design guidelines, CAD models etc. To further distinguish the terms “knowledge”, “information”, and “data” the definition by the VDI 5610 (Verein Deutscher Ingenieure 2009) which is aligned with other definitions found in literature, e.g. (Hicks et al. 2002; Stjepandić et al. 2015) is applied:

◊ “Data are objective facts, they cannot be interpreted without context and further backgrounds. They are to be taken as “raw material”.

- ◊ Information are structured data with relevance and purpose, which can be put into a context, categorized, calculated and corrected.
- ◊ Knowledge is linked information, which enables to draw comparisons, to establish links and to make decisions. "

In the context of design automation, various ways of structuring engineering knowledge are presented in literature. Baxter et al. (2007) distinguish process, task and product knowledge for formalization of design processes, automation of workflows and support of designers with formalized product knowledge based on a repository. The task knowledge refers to knowledge on algorithms and rules to update the product model according to the parametric input/output relations in the process model. Thus, the task knowledge builds the link to design automation methods with respect to KBE methods. J. H. Panchal et al. (2009) propose to distinguish procedural and declarative knowledge in design. The first refers to design process knowledge and the second to product knowledge with respect to parameters, interrelations etc. The latter is also used for the definition of task templates with well-defined inputs and outputs. To support formalization of these, Ming et al. (2015) propose an ontology-based approach to verify formalized knowledge based on rules. Recently, Ming, Zhenjun et al. (2018) present the corresponding implementation of the approach where three different use-cases of task template instantiation are distinguished: first, the instantiation of templates from the beginning by expert designers where the detailed content of the templates is defined such as the types of inputs needed for task formalization. Second, the adaptation of templates by senior designers is taken into account, e.g. for adaptation of the content of a task template with respect to changing goals. Third, the change of parameters within an already formalized design automation task template by a junior designers. Hence, it can be seen that the work aims at enabling designers to formalize tasks. However, the genericity of the approach does not take into account predefined templates that can be used to further guide designers when formalizing a design automation task through relations to the design process. Further, the approach focuses on tasks at a level of granularity that only take into account the changes of parameters, i.e. for parameter synthesis tasks. Thus, changes in the topology are not taken into account. Further, neither the usage of a standardized language that enables reuse of knowledge in a broader context for design automation task formalization is addressed.

Summarizing, it can be seen that different types of knowledge need to be considered for design automation task formalization. In this work, the application of templates to guide knowledge formalization and implicitly support knowledge externalization is further pursued. Template-based approaches are further extended by taking into account the different characteristics of state-of-the-art design automation opportunities. Whereas templates capture the relevant (declarative) product knowledge, different types of procedural knowledge need to be considered to enable design automation integration to design practice: first, with respect to the design process in which a design automation task is integrated and, second, with respect to the procedural knowledge required for solving a design automation task. The latter is inherent to the applied design automation method. To support integration of design automation task formalization to design practice, existing formalizations of knowledge need to be considered for design automation task formalization. Hence, alignment of the design automation task formalization with existing formalizations of knowledge is required. Thus, the application of a standardized language is required to rely on established interfaces and foster exchange of data.

2.3.2 Methodologies for Design Automation Task Formalization

Design automation methods are intended to address a specific type of design automation task rather than automation of the entire design process (Dym and Brown 2012). Yet, no comprehensive design automation task categorization that serves as an overview of these efforts and explicitly accounts for the different types of knowledge needed for formalization of a design automation task exist, cf. Section 2.1.1. Consequently, efforts towards a unified representation format that enables design automation method independent task formalization are missing, too (Colombo et al. 2014; Chakrabarti et al. 2011; Verhagen et al. 2012). Instead, various representation formats and knowledge acquisition methodologies that are tailored for one specific type of design automation methods can be found in literature. For example, the Methodology for Knowledge-based Engineering Applications (MOKA) was developed as a systematic for capturing product and process knowledge in engineering design for KBE implementation (Stokes and MOKA Consortium 2001). In analogy to systematics proposed for implementation of knowledge-based systems (KBS), e.g. CommonKADS (Schreiber et al. 1994) and the Model Based and Incremental Knowledge Engineering (MIKE) methodologies (Neubert 1993), MOKA features a two-step procedure for knowledge formalization. The first step captures expert knowledge based on informal documents and the second step aims at formalizing this knowledge within the MOKA modeling language that is built on UML. Due to the broad aim of automating repetitive tasks in product development, the MOKA does not provide any guidance with respect to what types of product and process knowledge should be captured for automation of a specific design automation task. Instead, it provides a language that enables comprehensive formalization and communication of expert knowledge and is aimed to support knowledge externalization by knowledge engineers (Verein Deutscher Ingenieure 2017). Hence, the methodology is not directed towards practitioners, but rather towards design automation experts or KBE developers, respectively.

2.3.3 SysML-based Design Automation Task Definition

Regarding the usage of a standardized formalization for design automation task definition, SysML has recently been addressed by multiple approaches identified in literature. SysML has evolved within the last ten years as a standardized modeling language to support model-based systems engineering (MBSE) (Friedenthal, Griego, and Sampson 2007; Friedenthal, Moore, and Steiner 2014). SysML as a model-based language aims to support communication and understanding of formalized knowledge (Friedenthal, Moore, and Steiner 2014). The language provides the full semantic foundation for documentation of system requirements, behavior, structure, and parametric relations. As a standardized language, SysML features reuse of models to avoid loose of knowledge between projects and reduce cost and risk in design (Beihoff et al. 2014). Approaches in the context of design automation feature systematics for definition of model-libraries for reuse (Kruse 2016; Wölkl 2012), formalization of parameter synthesis tasks (Shah et al. 2012), formalization of configuration tasks for KBE methods (Klein, Lützenberger, and Thoben 2015), formalization of simulation-based design tasks (Peak et al. 2007a) (Peak et al. 2007b), or neutral modeling of simulation models that can be then translated to the format of the desired simulation tool (Bock et al. 2017). Reason for the interest in SysML for design automation task formalization is the aspect of integration of formalizations to MBSE processes, and its means to support communication and understanding of formalized knowledge (Hotz et al. 2014). Next to definition of hierarchical relations within Block Definition Diagrams (BDD), it features graphical definition of relations between modeling elements within Parametric Diagrams (PAR) and Internal Block Diagrams (IBD). Previous efforts showed, that the modeling in UML/SysML enables automated transformation of the model to executable code based on mapping rules (Felfernig, Friedrich, and Jannach 2000) and stereotypes (Kerzhner 2012). Even though the introduced methods address design automation task definition in a neutral and standardized format, the approaches do not address the formalization of design automation tasks by designers. Instead of starting with the analysis of knowledge levels required for formalization and introducing systematics to guide and structure formalization, the approaches aim at extending the SysML to capture the semantics needed for automation based on a specific design automation method. Consequently, further abstraction is needed to capture the knowledge in SysML method independently and more structure needs to be provided to guide formalization of knowledge and enable practitioners to perform the knowledge acquisition and formalization themselves. In particular, the design automation task needs to be addressed from an engineering perspective rather than pure mathematical programming to enable design automation task formalization by designers.

2.4 Summary

In this section, the research gaps identified in the review of the relevant background for design automation task definition are summarized.

In a first step, the analysis of literature in the context of design automation task definition yields a taxonomy that contextualizes design automation and a design problem, design process, respectively, see Figure 2-1.

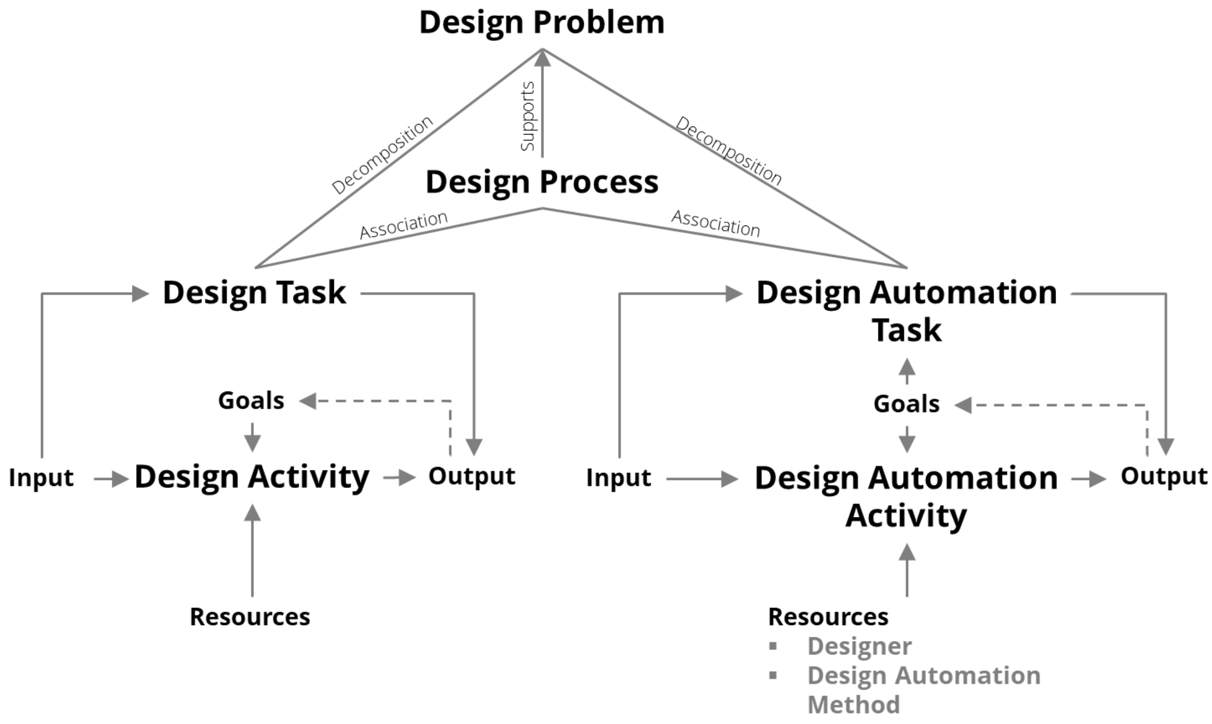


Figure 2-1: Clarification of vocabulary to put design automation into the context of design

Next, the literature review shows that a comprehensive design automation task categorization with respect to state-of-the-art design automation methods is missing. The categorization needs to contextualize a design automation task with the design process based on its characteristics that account for the relevant product knowledge as well as corresponding design states. Since each design automation task category features unique characteristics, the derivation of design automation task templates is enabled extending existing template-based approaches as indicated in Section 2.3.1. Thus, the categorization supports identification of design automation use cases as well as structuring of the product knowledge required for design automation task formalization.

With respect to design automation implementation in design practice, a lack of methods that enable identification of design automation use cases and integration of the related software applications and the technological environment and methods within the design process is recognized. ArchiMate as a standardized language features task precedence modeling of design processes in an integrated manner considering also the tools and technology perspective of design. Further, as indicated above, design automation task templates that account for the design process perspective can be derived. Implementation of these templates in design process models enables specification of interfaces required for integration of design automation to design practice. Hence, a method for identification of design automation use cases and design automation integration can be derived.

Regarding the impact estimation of design automation, a need for a method enabling impact estimation as well as related metrics is identified. Regarding the latter, the review of related literature shows that existing standards from software quality assessment as well as design automation drivers that reflect the motivational aspects of design automation implementation can be used to organize a metrics system for impact estimation of design automation. With respect to the actual derivation of metrics for impact estimation of design automation, best practice methods for metrics derivation pursue a top-down approach and start with the definition of goals before the actual definition of metrics for quantification of the goals' achievement. Hence, the same strategy will be pursued in this work where a top-down approach is used for the derivation of metrics based on the analysis of design processes in practice and identification of design automation goals. More particularly, design process value stream mapping and design process failure modes effect analysis (dpFMEA) will be conducted to identify waste drivers and potential failure modes in design, respectively. The dpFMEA serves for identification of potential failure modes but also for analysis of related causes and effects. Taking into account that the removal of causes refers to the design automation goals and information on the effects can be used as questions to guide metrics definition, a GQM based method for derivation of metrics is proposed in this work. Thereby the proposed methodology contextualizes design automation and design practice based on potential failure modes reflecting the designers' needs for support. Repeated application of the method to different industrial cases leads to the establishment of a design automation metrics system that can be reused for a method for estimation of the impact of design automation. In particular, metrics can be linked to specific design automation task templates to account for the characteristics of specific design automation tasks. Implementation of design automation task templates in design process models makes a list of metrics readily available that can be filtered according to the characteristics of the use case. Thus, comprehensive assessment of the impact of design automation on design practice that considers both the design process perspective as well as the respective task is enabled.

Finally, related literature on knowledge formalization reveals that only few methods are available that aim at enabling design automation task formalization by designers. The review showed that design automation task templates are a means to foster design automation task formalization by designers and the application of UML based languages supports communication of knowledge. Yet, existing methods for design automation task formalization are restricted to design automation tasks focusing on synthesis of parameters and are based on case- and design automation method-specific formalizations. Hence, in this work a method that is based on the application of SysML as a standardized language in combination with design automation task templates to guide knowledge formalization is developed to enable designers to formalize a design automation task themselves.

3 Industrial Cases

Following the DRM (Blessing and Chakrabarti 2009), two different aspects of evaluation are considered in this thesis: application evaluation for demonstrating applicability of the proposed methods and success evaluation to assess the usefulness of obtained results. In total, three industrial cases are used for evaluation of the methods.

The first industrial case focuses on structural analysis of crane structures, which includes pre- and post-processing of simulations as well as adaption of dimensioning of parameters (Case 1 – Structural Analysis). The design of cranes is strongly regulated; hence, a large set of use cases needs to be validated using structural simulations, which makes it a tedious and complex task that requires deep expert skills. The investigated process can be considered of relatively routine character and the degrees of freedom for the designer are limited to a set of dimensioning parameters. Figure 3-1 shows a schematic drawing of a box-type boom that refers to a commonly used structure for cranes.

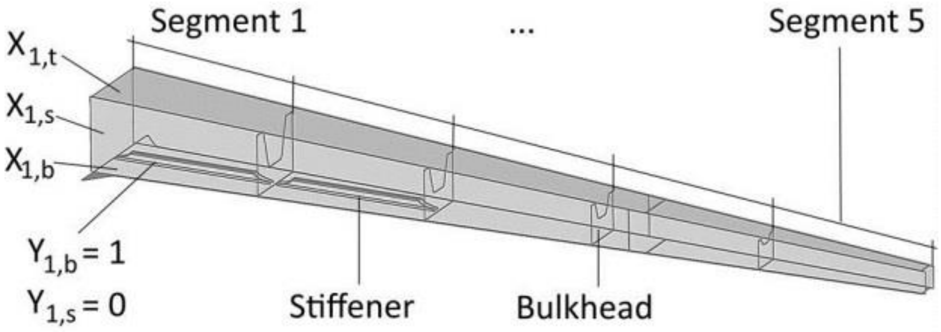


Figure 3-1: Schematic drawing of a box-type boom structure indicating the dimensioning parameters.

The second industrial case focuses on the design of hydraulic systems for heavy construction machinery (Case 2 – Hydraulic Systems). This includes selection of concepts for realizing the hydraulic circuits for a given set of requirements and boundary conditions as well as selecting appropriate components and parameters. Thus, for this case the design space is less strict and involves design activities that require creativity and experience to identify appropriate solutions, in particular, for concept identification. Figure 3-2 illustrates the layout of a closed circuit hydraulic loop.

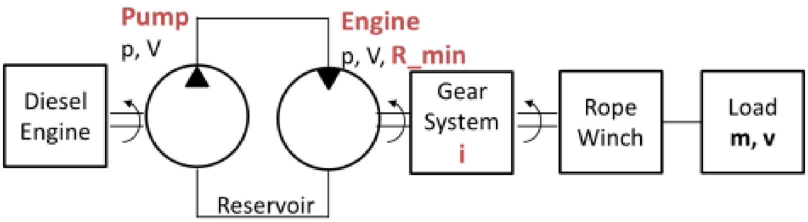


Figure 3-2: Closed circuit hydraulic loop indicating the design parameters in red.

The third industrial case addresses the definition of layouts for design of hydraulic units based on requirements stemming from the sales department (Case 3 – Hydraulic Units). The investigated process spans from sales to detail design of hydraulic aggregates and its performance is mainly determined by the process lead time. For a given selection of components and performance specifications, a geometric arrangement of components on a restricted space needs to be defined. Thus, the design tasks involve embodiment design activities that require expert knowledge for identification of appropriate solutions, in particular, regarding concepts for arrangement of components to fit the functional and geometrical requirements. Figure 3-3 shows a CAD model of a hydraulic unit.

The industrial cases were conducted with multinational companies and the contributing designers are located in Austria and Germany.

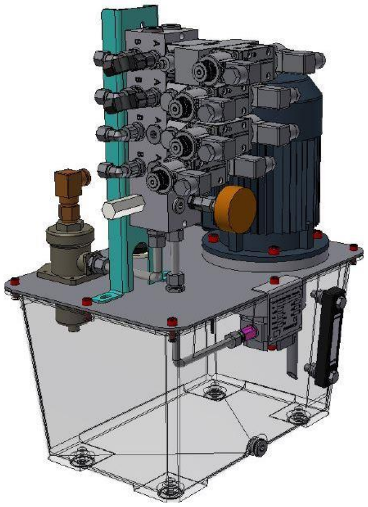


Figure 3-3: CAD model of hydraulic unit (Kunz Maschinenbau, 2019).

4 Methodology for Design Automation Task Definition

In response to the need highlighted in Section 2.4, this section presents the methodology for design automation task definition. It aims to enable implementation of design automation in design practice according to designers' needs and focuses on design automation methods from CDS and KBE. First, Section 4.1 introduces the methodology, its basic characteristics, the four steps and supporting methods that constitute the methodology as well as the design automation task categorization that builds the basis of the methodology. For each method as well as the design automation task categorization, the relation to the research questions is indicated as well as the link to Sections five to nine where the supporting methods as well as the design automation task categorization are each detailed. Following this, Section 4.2 contextualizes the methodology for design automation task definition based on a meta-model in terms of the design process, supporting tools and technology and their relation to motivational aspects of design automation implementation. Also, the meta-model features the link of the design process to the product knowledge formalization necessary for design automation task formalization. Hence, a meta-model is established that lays out the vocabulary and interrelations for reasoning within the scope of the proposed methodology. Finally, Section 4.3 addresses the industrial evaluation of the proposed methodology and introduces the corresponding industrial cases.

4.1 Overview of Methodology for Design Automation Task Definition

In Figure 4-1, the intended scenario for application of the methodology for design automation task definition in industrial practice is presented: It starts with selection of components or systems for which support in design is desired and ends with the implemented and integrated design automation solution. The four-step procedure of the methodology is indicated as well as its dependence on the supporting methods. The developed methodology is to be used within collaborative workshops with practitioners to enable development of design automation tailored to the needs of practitioners and thereby increase acceptance. Further, the application of the methodology is intended for design processes that are conceptually well established and exhibit a relatively routine character with respect to the structure of the process, which is often the case in industrial practice (Wynn and Clarkson 2018). In context of this work, this guarantees an adequate design process maturity (CMMI 2010) that permits task precedence and dependency-based modeling of design processes. Although the focus on this type of design processes is a limitation on the range of all design process types that might be considered, the relatively routine design processes can still contain iterations (Wynn and Clarkson 2018) and characteristics of innovative and creative designing (Gero 2000). This allows generalizing the results of this work for design automation tasks beyond the automation of standard, routine design tasks.

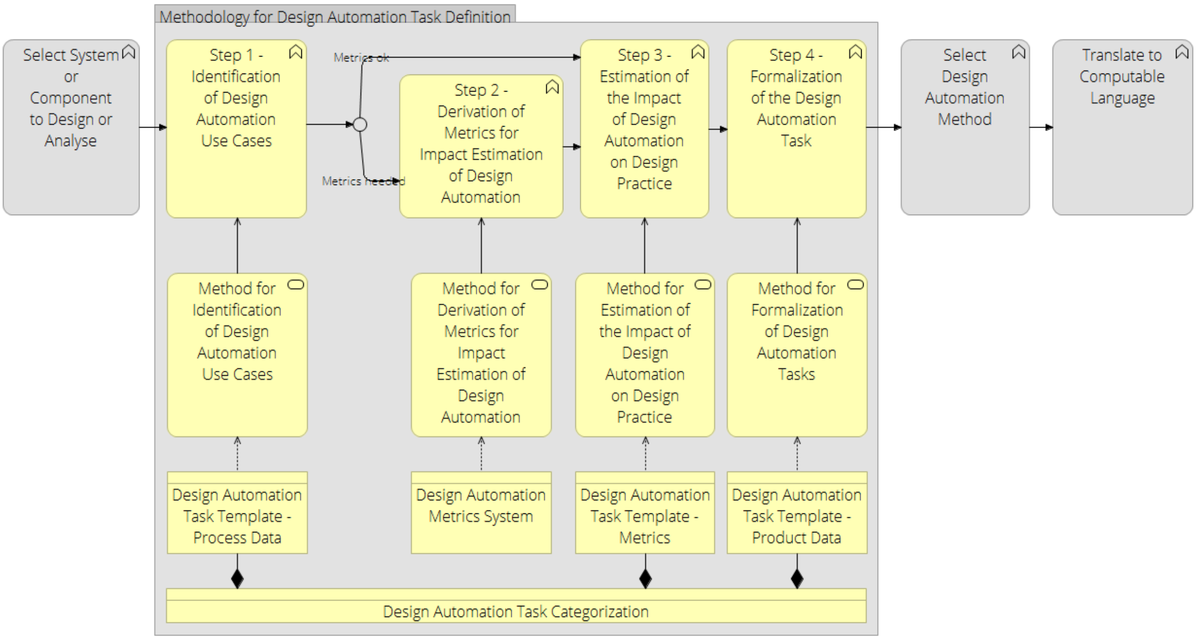


Figure 4-1: Overview of the four-step methodology for design automation task definition as well as supporting methods and information developed in this work. Steps that are outside the visual group element are considered out of scope of this thesis.

4.1.1 The Design Automation Task Categorization

It can be seen within Figure 4-1 that the methodology for design automation task definition builds on the **design automation task categorization** and related design automation task templates. The design automation task categorization is based on a state-of-the-art review of design automation methods from the research fields KBE and CDS. The explicit listing of design automation task characteristics that are specific to each category summarizes the opportunities state-of-the-art design automation offers from a design automation task perspective. For each category, three templates can be derived to account for design automation task definition from multiple perspectives:

- A design process perspective indicating what sort of data is needed for design automation implementation and how to integrate design automation, design activities and related tools and technologies within the design process.
- A motivational perspective to relate metrics for impact estimation of design automation to specific design automation tasks.
- A product knowledge perspective to put into context the information that is obtained from the design process analysis and the corresponding product knowledge needed for design automation task formalization.

The pursued approach for the **derivation of the design automation task categorization** is presented in Section 5 and aims to answer **research question one**: "What are the characteristics of design automation tasks that enable identification of design automation opportunities in design practice?" as well as **consolidation** of the **heterogeneous research field design** automation with respect to **KBE and CDS**.

4.1.2 Step 1 – Identification of Design Automation Use Cases

After selection of a system or component for which design performance improvement is desired, the very first step of the 4-step methodology comprises the identification of design automation use cases for a given design process. This step is supported by the **method for identification of design automation use cases** and is based on an integrated modeling and analysis of the design process (Step 1). Particular focus is put on design process decomposition including the supporting tools and technologies to comprehensively analyze the design process. Regarding integration, design automation task templates are used for identification of relevant product knowledge, its specific formalizations as well as the supporting tools that are relevant for design automation implementation. The details of process modeling, decomposition and mapping of templates are presented in Section 6, where the method for identification of design automation use cases is introduced in order to address the **research questions two and three**: "What knowledge is necessary for the identification of design automation use cases in design practice?" and "How can we best integrate design automation in industrial practice?"

4.1.3 Step 2 – Derivation of Metrics for Impact Estimation of Design Automation

For identified design automation use cases, metrics for impact estimation of design automation are derived (Step 2) based on the **method for derivation of metrics for impact estimation of design automation** that is presented in Section 7. The proposed method builds on failure modes analysis in the design process. This is based on an assumption that if design automation is used within a design process some of the failure mode causes are removed and effects are mitigated. The formulation of design automation goals as removal of failure modes' causes enables the top-down method for derivation of metrics based on the GQM paradigm. To support identification of appropriate metrics, the design automation metrics system that is introduced in Section 2.2 can be reused as a basis for identification of appropriate metrics in a given context, e.g. the duration of a specific iteration within the process to assess productivity or the engineering first pass yield to measure the error rate. The method for derivation of metrics for impact estimation of design automation aims to answer **research question four**: "What are appropriate metrics to assess the impact of design automation on design performance?"

4.1.4 Step 3 – Estimation of the Impact of Design Automation on Design Practice

The results yielded from successive application of the method for derivation of metrics can be used to enrich the design automation metrics system to enable reuse. Further, metrics that account for the motivational aspects for design automation implementation according to a specific design automation task category can be associated to the corresponding design automation task template. Thus, the method for estimation of impact of design automation on design practice can be potentially applied without the need for derivation of metrics. Instead, design automation task templates that account for the motivational aspects can be integrated into the design process model and readily provide a list of relevant metrics for potential estimation of design automation. Thus, metrics can be filtered according to the specific interests of a company and associated to the addressed artifacts, e.g. lead time assessment of a specific design activity. Estimation of the metrics values establishes the basis for making an informed decision as to whether the implementation of design automation is desirable or not. Section 8 details the **method for impact estimation of design automation on design automation** that addresses **research question five**: “How can the impact of design automation on design practice be estimated prior to the implementation?”

4.1.5 Step 4 – Formalization of the Design Automation Task

Following a positive decision with respect to design automation implementation, Section 9 presents the **method for design automation task formalization**. The method is based on design automation task templates that are used for guiding designers in knowledge formalization and support modularization of the formalized knowledge. The SysML modeling language is applied for graphical formalization of product knowledge based on a standardized language. This addresses the need for a method enabling design automation task formalization by designers to mitigate black-box perception of design automation in industry and reduce efforts for knowledge acquisition and formalization. The systematic application of a neutral and exchangeable format, the strict separation of design automation task formalization and mathematical programming as well as the structuring of knowledge based on design automation task templates address **research questions six and seven**, i.e. “In what aspects does the usage of a graphical modeling language support designers to formalize design automation task themselves?” and “How can the completeness of a task be assessed in order to support designers when formalizing a design automation task?”

4.1.6 Steps not covered in the scope of this work

After successfully conducting the 4 steps of the methodology for design automation task definition, an appropriate design automation method needs to be selected. This enables (automated) translation of the task formalization to the formalization specific to the selected design automation method, e.g. graphs. Thus, the transformation of a design automation task to a mathematical problem can be conducted. Finally, after successful transformation of the design automation task definition to executable code, the design task is automated. Yet, the focus of the thesis is on the methodology for design automation task definition which covers the steps prior to implementation and is introduced in detail in the subsequent sections. For the sake of demonstration of the scenario, a proof of concept implementation for automated translation of a design automation task formalization to executable code was conducted. Thereby, the principles of modularization inherent to the templates were exploited to structure and guide code generation.

4.2 Meta-Model for Integrated Assessment of Design Processes

Figure 4-2 contextualizes the design process, including the supporting tools and technologies as well as motivational aspects for design automation implementation based on the ArchiMate 3.0 (The Open Group 2016) language. It is shown how the proposed methodology relates failure modes, causes and effects in design processes with metrics, drivers, targets and design automation goals. This is necessary for estimation of the impact of design automation on design practice and communication of the value of design automation. All elements related to the motivational aspects for design automation implementation are depicted in purple and captured within the box titled "DA Metrics Meta-model". The design workflow, corresponding design activities, actors in the design process, inputs and outputs are captured in the ArchiMate's business layer, which is depicted in yellow. Applications of supporting design process methods and corresponding data objects are captured in the Application Layer and are indicated by blue artefacts. The corresponding realizations as design tools, related technologies and formalizations are represented within the technology layer that is characterized by green elements. It has to be noted, that the arrows from supporting tools and technologies are directed towards the objects of the business layer to indicate the supporting character. For the case of information backflow, this is captured in the related input and output data objects.

To relate formalized product knowledge with the design automation task formalization, it is shown how the identified data objects can be mapped to SysML diagrams to support alignment, reuse and synchronisation of knowledge. Hence, the concept of linking the ArchiMate language with the SysML permits to establish dependency and traceability relationships between the two. In detail, Block Definition Diagrams (BDD), Internal Block Diagrams (IBD) as well as corresponding Parametric Diagrams (PAR) are used for formalization of the design automation task knowledge with respect to product knowledge, such as product architectures, model libraries, rules etc.

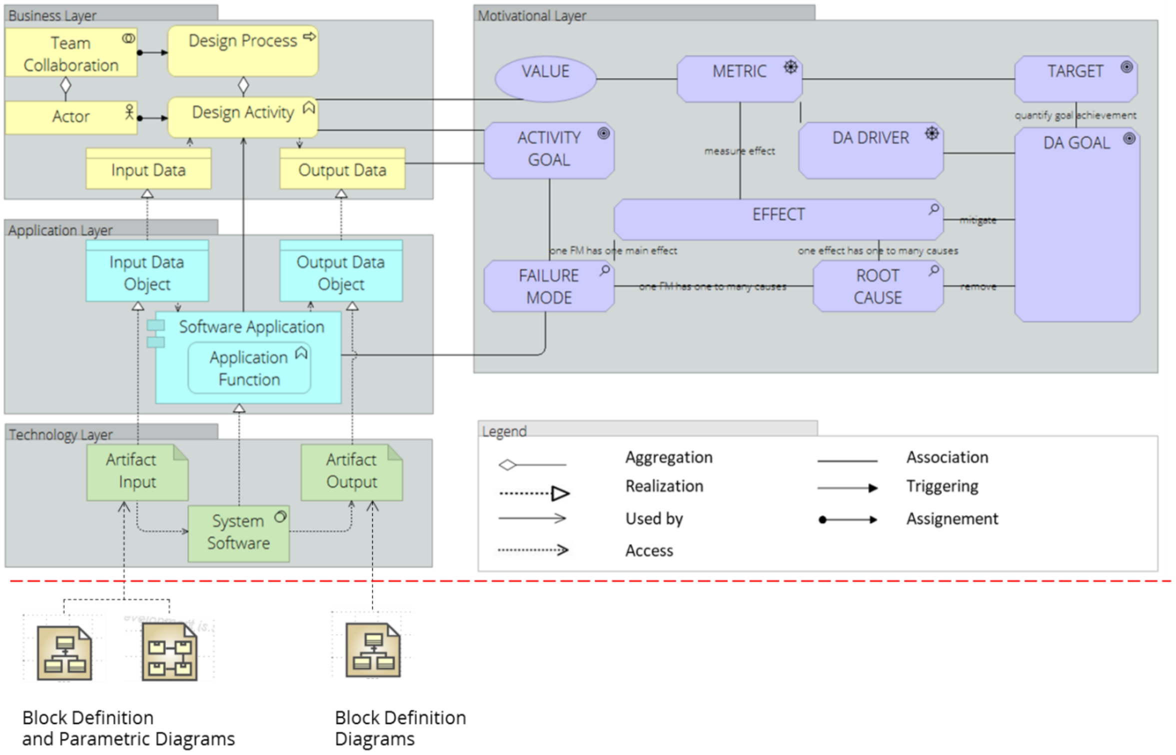


Figure 4-2: Meta-model of the proposed methodology relating the motivational aspects for design automation implementation (purple) to the design process (yellow), applications of design process support methods (blue), their realizations as design tools and related technologies (green) and the formalization of product knowledge in SysML block definition and parametric diagrams.

4.3 Evaluation of Methodology for Design Automation Task Definition

Due to the bottom-up character of the proposed methodology and its aggregation of multiple sub-methods, successive evaluation of the sub-methods is conducted to evaluate each method on its own and based on the results deduce the applicability of the overall methodology presented in this thesis.

Table 4-1 indicates the application scenarios of each industrial case, i.e. which methods were applied. The methodology for design automation task definition was completely applied for Case 2 including automated transformation of the design automation task definition to a computable formalization. This allows evaluation of the methodology as a whole by demonstrating its feasibility for design practitioners.

Table 4-1: overview of use cases and applied methods

Use Case	Applied Method
Case 1 – Structural Analysis	Method for Identification of Design Automation Use Cases, Method for Derivation of Metrics, Method for Estimation of the Impact of Design Automation on Design Practice
Case 2 – Hydraulic Systems	Methodology for Design Automation Task Definition
Case 3 – Hydraulic Units	Method for Identification of Design Automation Use Cases

5 Design Automation Task Categorization and Templates

In response to the need for a comprehensive overview of design automation opportunities, this section introduces the pursued research methodology for categorizing design automation tasks as well as the corresponding results yielded from analysis of design automation methods. Section 5.1 builds on the findings presented in Section 2, and establishes criteria for analysis of design automation methods based on knowledge levels needed for design automation task definition, as well as knowledge formalization and reasoning techniques. After clustering of the results that are yielded from analysis of 77 design automation methods from the fields of KBE and CDS, the design automation task categorization is derived in Section 5.2. Building on these results, Section 5.3 introduces the derivation of design automation task templates according to the viewpoints indicated in Section 4.1, i.e. design process data, motivational aspects as well as product data. The templates build on the criteria presented in Section 5.1 as well as the meta-model defined in Figure 4-2. Section 5.4 discusses the implications of attained results on design practice and research including a consolidation of the research fields CDS and KBE. Finally, Section 5.5 presents a summary section where the relation to research question one is indicated and corresponding contributions are highlighted.

5.1 Criteria for Systematic Analysis of Design Automation Methods with respect to Design Automation Tasks

Section 2 shows that assessment of knowledge levels with respect to inputs, outputs as well as goals is required for the comprehensive analysis of design automation tasks. More particularly, in the context of design automation tasks, in- and output knowledge correspond to the product knowledge available or desired, respectively. The goals are investigated to account for the knowledge needed to control and guide the design space exploration for attaining the desired output. To this end, the purpose of the design automation task, addressed system levels as well as the requirements, constraints and objectives are analyzed. Domain as a type of goal refers to restrictions due to discipline characteristics such as availability of analysis methods, types of standards and regulations to be considered (Pahl et al. 2007). To ensure the generality and domain independence of the categorization, this aspect of design automation task formalization is not further considered in the approach pursued in this work. Lastly, to account for the characteristics of design automation methods that are available for automation of the design tasks, the knowledge representations as well as reasoning techniques are examined. Figure 5-1 shows how the knowledge levels on input, output and goals of a design automation tasks as well as the resources with respect to design automation methods can be put into context with a design automation activity “Execute Design Automation Task”.

The following subsections 5.1.1 – 5.1.3 gradually refine the stated criteria for systematic analysis of design automation tasks, i.e. inputs, outputs, goals and design automation methods. Additionally, for each criterion a set of normalized instance values is introduced that is summarized in Figure 5-2.

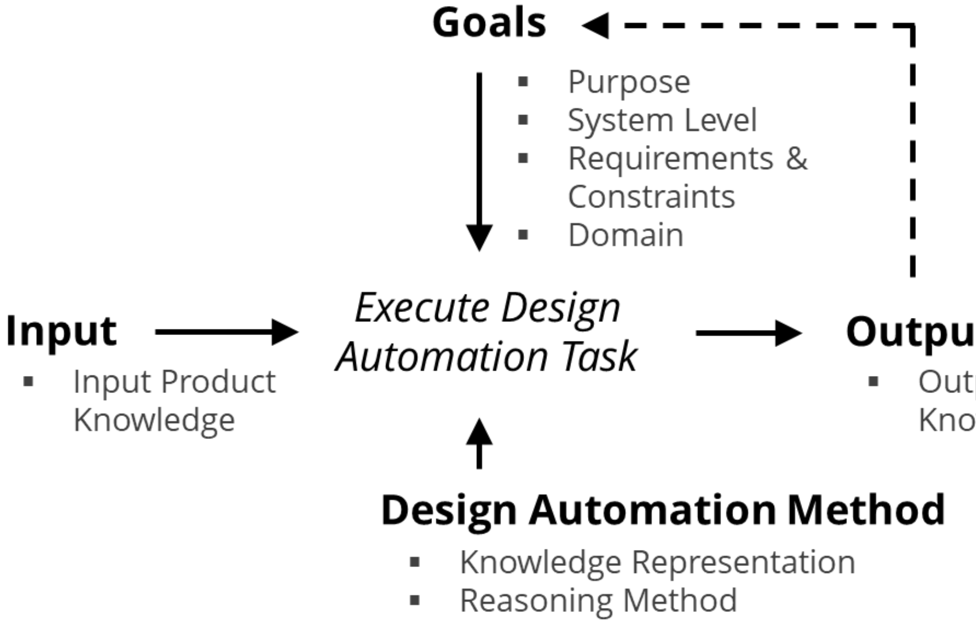


Figure 5-1: Scheme of explicit knowledge needed for design automation task definition

5.1.1 Input / Output

In- and output of a design automation task are characterized by the product knowledge that describes the state of the initial and final product definition, respectively. The type of in- and output models are strongly linked to the design process stage and are represented as:

- ◊ *Transformation processes (process),* i.e. how the design is transforming inputs to outputs, e.g. through manufacturing processes (Hubka and Eder 1982),
- ◊ *Main function* or *functional models (function)*;
- ◊ *Product architectures (prod.arch.)* that describe the composition of a design without consideration of spatial attributes;
- ◊ *Geometric models (geometry)*, mostly represented by CAD-models but not limited to. Geometric models also account for general arrangements of system units and layouts.

Knowledge of the output can be further enriched by information regarding the *Bills of Materials* (*BoM*), *cost estimation (cost est.)*, *simulation models (simulation);* or *standardized models (stan. Model*), i.e. a product platform.

Related to the input, additional knowledge to automate the design task is required whereby the following aspects are differentiated:

- ◊ *Component libraries (comp.lib.)*, which refer to collections of building blocks such as functional libraries, CAD-libraries, simulation model libraries etc.;
- ◊ *Interconnections (intercon.)*, i.e. how components contained in a component library can be related to generate designs, e.g. dependencies of parameters and interface specifications;
- ◊ *Modifications (modific.)*, that account for the degrees of freedom of a design automation task and highlight potential changes that can be applied to a model, e.g. the variables and parameters of a design;
- ◊ *Mappings (mapp.)*, regarding mapping between different abstraction levels; and
- ◊ *Performance evaluation (eval.alg.)*, methods and knowledge for determining the product performance according to the available design criteria given the design stage and corresponding abstraction level.

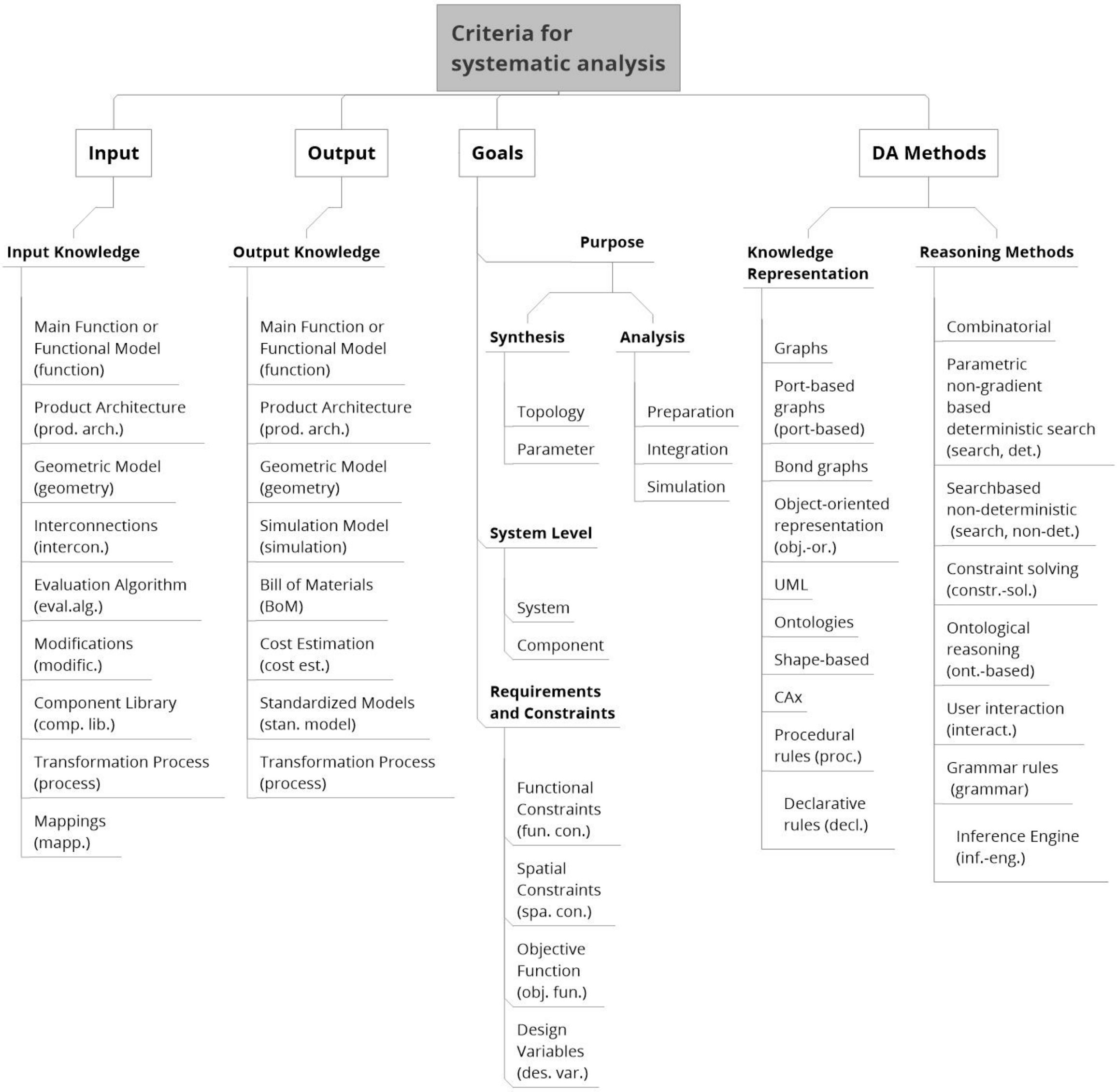


Figure 5-2: Tree representation of criteria. Leafs indicate the corresponding instance values for systematic analysis of design automation methods. Abbreviations of instance values are put in brackets.

5.1.2 Goals

Regarding goal formalization, the purpose, the addressed system level as well as requirements and constraints will be analyzed.

The main categories introduced by Sim and Duffy (2003) will be used as criteria for analysis of the **purpose** of a design task, i.e. whether the intention is to define or evaluate a design. Thus, it is distinguished whether a task aims at *design synthesis* or *design analysis*. Generally, formal synthesis refers to “algorithmic creation of designs” (Cagan et al. 2005) on any abstraction level, and regarding the present paper, can be performed with respect to synthesis (or generation) of *topologies* on conceptual and embodiment representation levels and associated *parameters*. With respect to the latter, parameters of the current design state or transformation processes are gradually refined by means of parameter instantiation, potentially based on available design templates for standardized products. With respect to analysis, model *preparation*, e.g. geometry reduction, segmentation, meshing; *integration*, assembling/integrating components of existing simulation models; and, *simulation*, the automated definition of simulation parameters and actual execution of the simulation are distinguished.

Regarding the addressed **system level**, it is analyzed whether an entire *system*, i.e. assemblies or a single *component* is in focus of the design task in order to attain additional criteria for assessing granularity of tasks.

Lastly, in order to fully specify a design task, **requirements and constraints** need to be defined that can be represented by *functional constraints (fun.con.)*, i.e. the performance specification; *spatial constraints (spa.con.)*, referring to restrictions regarding topologies, parameters or spatial constraints in general; or general *design variables (des.var.)*, referring to few characteristic parameters of products that implicitly contain design choices (Moullec et al. 2013), e.g. performance or geometric attributes of components. In case optimized designs are targeted, an *objective function (obj.fun.)* needs to be defined.

5.1.3 Design Automation Methods

In the following, categories of knowledge representations as well as reasoning methods are introduced in order to account for method specific characteristics. Whereas Section 5.1.3.1 focusses on the representations used for declarative modeling of input knowledge as well as goals, Section 5.1.3.2 lists the main paradigms used for reasoning according to given inputs and goals, i.e. for control and guidance of the solution space exploration.

5.1.3.1 Knowledge Representation

In order to be able to derive an interoperable formalism for declarative design task formalization, formalisms and representations used for existing methods need to be analyzed. Knowledge representations specific to design automation methods depend closely on the reasoning method applied. Thus, the following different product knowledge representation techniques are defined:

- *Graphs* are used in the context of design automation as a basis for formalization of components and interrelations. Graphs are often used to model networks for representation of uncertainties or constraint networks and are especially useful when both formal as well as visual representation is requested. With regards to product development and design automation methods investigated within this paper, the following specific types of graphs will be distinguished:
    - *Port-based graphs (port-based)*, which are simple flow graphs used to model functional and component structures, and,
    - *Bond graphs*, for modeling the dynamics of systems based on generic models of components with various types of connection possibilities.
- *Object-orientated representation (obj.-or.)* denotes the application of object-oriented modeling for achieving generality of knowledge by structuring it using concepts such as inheritance. Standardized methods for graphical modeling of object-oriented systems exist, for example *UML*.
- *Ontologies* are a means for representing the knowledge of the domain of interest with explicit formalization of the semantic relations of knowledge objects.
- For certain design automation methods, reasoning with shapes is required. Hence, *shape based* representations are applied. For most methods considered in this paper, *CAx* is then used for representation of geometries. However, more abstract representations can also be applied (Chakrabarti et al. 2011).
- For routine design tasks with well formalized processes and activities, often *procedural rules* are applied: a defined sequence of transformation rules incorporating knowledge on how to transform the input to the output for a given goal, are applied. More generally, a set of *declarative rules* can be used in combination with an inference engine for transformation of knowledge.

5.1.3.2 Reasoning Methods

As stated by Brown and Chandrasekaran (1990), a crucial measure for categorization of design tasks is the availability of knowledge on how to solve a task. In this study, the knowledge of reasoning techniques is used to indicate opportunities with respect to available design automation methods.

Following paradigms for guiding the search for solutions according to given inputs and goals often appear for state-of-the-art design automation methods considered in this paper:

- Combinatorial methods including enumeration *(combinatorial)*
- Parametric, gradient-based deterministic methods such as Newton's method,
- Parametric, non-gradient based deterministic search methods, such as Simplex Algorithm *(search, det.),*
- Search based non-deterministic methods including meta-heuristics *(search, non-det.),*
- *constraint solving (constr.-sol.),*
- *ontological reasoning (ont.-based)*, and
- *user interaction (interact.)* for semi-automated approaches,
- *inference engine (inf.-eng.)* in order to guide rule application, e.g. based on forward- or backward-chaining.

For non-routine design tasks and exploration of solution spaces, *grammar rules (grammar)* can be formalized where a rule set in combination with a specific vocabulary determines how the design space can be altered (Chakrabarti et al. 2011). The application of grammar rules itself can be guided by general search-strategies, optimization algorithms etc. Hence, a combination of artificial intelligence and operations research is yielded (Cagan, Grossmann, and Hooker 1997).

5.1.4 Criteria and Instance Values for Systematic Analysis

Figure 5-2 summarizes the refinements and established instance values of criteria for systematic analysis of design automation tasks. The leaves of the tree visualization list both instance values as well as corresponding abbreviations in brackets.

5.2 Design Automation Task Categorization

For derivation of a comprehensive design automation task categorization, this section provides a systematic analysis of state-of-the-art design automation methods according to previously defined criteria as defined within Figure 5-2. Results are structured according to the model introduced in Figure 5-1, i.e. inputs, outputs, goals and design automation methods. With regards to literature selection, a particular focus is put on recent research results describing methods and tools of practical applicability. Based on the analysis of types of knowledge needed for formalization of design automation tasks according to method specific implementations, a generic categorization of design automation tasks is derived. The categorization is supposed to describe the knowledge needed for design automation in a method independent, however, design automation task category specific manner.

5.2.1 Analysis of State-of-the-Art Design Automation

The following Table 5-1 shows the results of the detailed investigation of design automation methods according to the criteria as defined in Figure 5-2. In order to further illustrate the broad applicability of design automation over multiple domains and reduce the level of abstraction, case studies of the listed approaches are indicated. It has to be noted that not all case studies have been applied in industrial environments. For some design automation methods, test cases that are considered to have a high industrial relevance are selected for academic validation in order to demonstrate the method’s applicability, e.g. (Münzer and Shea 2015). Table 5-1 is organized to show the design automation methods according to shared characteristics of addressed design automation tasks, in particular, the output and whether a synthesis or analysis task is addressed. Consequently, a preliminary design automation task categorization is achieved. These results will be used in the following for derivation of a general design automation task categorization.

5.2.2 Design Automation Task Categorization

Based on the findings of Table 5-1, a clustering of design automation tasks based on shared characteristics with respect to output and purpose of a task is conducted. Therefore, knowledge on inputs and goals is integrated among methods for each design automation task category. This enables to account for all type of input knowledge which is potentially hidden in method specific formalizations. Table 5-2 summarizes the design automation tasks and depicts the knowledge needed for formalization and available design automation methods according to the criteria presented in Figure 5-2. Figure 5-3 illustrates the design automation task categorization by means of a crane system design case: design states as well as different levels of granularity are shown in the context of a generic design process according to Pahl et al. (2007).

Table 5-1: Review of state-of-the-art design automation methods according to the metrics defined in Section 4.1. Column 1 provides a brief description of derived clusters of addressed design automation tasks. "[…]" is used as a means to reuse the naming from the row above. "top." refers to topology, "par." parameters, "opt." optimization.

Design Automation Task (*derived*)	**Reference**	**Input**									**Output**								**Goals**										**Design Automation Methods**																		**Case Study**
																			Syn.		**Analysis**				**Req. & Con.**				**Knowledge Representation**										**Reasoning Technique**								
		function	prod.arch.	geometry	process	comp.lib.	intercon.	modific.	mapp.	eval.alg.	function	prod.arch.	geometry	simulation	process	BoM	cost est.	stan. model	topology	parameter	simplification	integration	simulation	Level	fun.con.	spa.con.	obj.fun.	des.var.	grpah	port-based	bond graph	obj.-or.	UML based	ontology	shape based	CAx	proc.	decl.	Combinatorial	search, det.	search, non-det.	constr.-sol.	ont.-based	interact.	inf.-eng.	grammar	
Functional Synthesis	Jin and Li, 2007	x				x	x		x		x								x						x		x		x												x					x	personal transporter
	Yoshioka et al., 2004	x				x	x		x		x								x															x									x				laser stero-lithography
	Kitamura and Mizoguchi, 2004			x		x	x		x		x								x															x									x				wire saw
	Helms and Shea, 2012	x				x	x	x	x		x	x							x						x				x			x								x						x	hybrid powertrain architecture
Product Architecture Synthesis (Top. & Par.)		x				x	x	x	x		x	x							x	x				s	x				x			x								x						x	
	Campbell et al., 1999	x				x	x	x	x			x								x				s	x						x									x							weighing machine
	Moullec et al., 2013					x	x	x	x			x								x				s	x	x																x					radar antenna
	Komoto and Tomiyama, 2012					x	x	x	x		x	x								x				s	x									x									x				standard printer design
[…] incl. Opt.	Königseder et al., 2013					x	x					x							x	x				s	x	x	x		x											x	x					x	photovoltaic arrays
	Bayrak et al., 2016					x	x			x		x							x	x				s	x	x	x				x									x							hybrid powertrain
[…] incl. Opt. & Analysis	Münzer et al., 2015	x				x	x	x	x			x		x					x	x		x	x	s	x	x				x	x	x						x			x	x					hybrid powertrain
	Canedo et al., 2013	x				x	x	x	x			x		x					x	x		x	x	s								x					x										popcorn machine
	Canedo and Richter, 2014	x				x	x	x	x			x		x					x	x		x	x	s								x					x										hybrid powertrain architecture
Product Architecture Analysis	Neyrinck et al. , 2015;			x		x	x		x			x		x								x	x	s	x							x				x	x										machining system
Product Architecture Synthesis (Top.)	Kurtoglu and Campbell, 2009	x				x	x		x			x							x					s	x				x											x						x	wall-climber toy
	Bryant et al. , 2005	x				x	x	x	x			x							x	x				s								x							x								box-labeling device
	Schmidt and Cagan, 1997	x				x	x		x			x												s			x		x												x					x	meccano toys
	Schmidt et al., 2000		x					x				x												s					x										x							x	epicyclic gear trains
	Wyatt et al., 2012		x			x	x					x												s		x																x					vacuum cleaner
	Münzer et al., 2013					x	x					x												s	x	x				x												x					hybrid powertrain
Product Architecture Synthesis (Par) incl. Opt.	Hutcheson et al., 2006	x	x					x		x		x								x				s			x					x									x						power-head for tools
	Wu et al., 2008		x					x		x		x								x				s			x				x										x						elec.-mech.- design
Spatial Product Architecture Synthesis (Top. & Par.)	Lin et al., 2009			x			x	x		x			x						x	x				s	x	x	x								x						x					x	gear trains
	Heisserman et al., 2004			x			x	x					x						x	x				s		x									x									x		x	piping for airplanes
	Boehnke et al., 2009		x			x		x					x						x	x				s				x					x				x										airplane topologies

Table 5-1 continued

Design Automation Task (*derived*)	Reference	Input									Output								Goals										Design Automation Methods																		Case Study
																			Syn.		Analysis				Req. & Con.				Knowledge Representation										Reasoning Technique								
		function	prod.arch.	geometry	process	comp.lib.	intercon.	modific.	mapp.	eval.alg.	function	prod.arch.	geometry	simulation	process	BoM	cost est.	stan. model	topology	parameter	simplification	integration	simulation	Level	fun.con.	spa.con.	obj.fun.	des.var.	grpah	port-based	bond graph	obj.-or.	UML based	ontology	shape based	CAx	proc.	decl.	Combinatorial	search, det.	search, non-det.	constr.-sol.	ont.-based	interact.	inf.-eng.	grammar	
Spatial Product Architecture Synthesis (Par.)	Shah et al. , 2012			x		x	x	x	x				x							x				s		x	x						x									x					hydraulic log splitter
[…] incl. Opt.	Coorey and Jupp, 2014			x				x					x							x			x	s		x	x					x					x				x						floor layout design
	Yin and Cagan, 2000			x		x	x	x					x							x				s		x	x					x								x							heat pump layout
[…] incl. analysis	Van der Laan and Van Tooren, 2005					x		x	x				x	x						x	x	x	x	s				x				x				x	x										aircraft movable
	Cooper and La Rocca, 2007					x		x	x				x	x						x	x	x	x	s				x				x				x		x							x		wind turbine
	La Rocca and Van Tooren, 2010					x		x	x				x	x						x	x	x	x	s				x				x				x		x							x		aircraft concepts
	Johansson, 2012; Johansson and Elgh, 2013					x		x	x				x	x						x	x	x	x	s				x				x				x		x							x		car rack fixtures
Spatial Component Synthesis (Top. & Par.)	Hoisl and Shea, 2013			x				x					x						x	x				c											x									x		x	robot arm concepts
	You et al., 2011; Kulon et al., 2006			x		x	x	x					x						x	x				c				x				x				x	x			x							car panel die face; metal forging parts
	Tsai et al., 2010			x		x	x	x					x						x	x				c				x				x				x	x			x							die design
Transformation Process Planning based on Geometry				x		x	x	x							x				x	x				c				x				x				x	x			x							
	Chulvi et al., 2007					x	x		x						x					x			x	c	x	x										x		x							x		skull implant
Spatial Component Synthesis (top.) and Analysis						x	x		x				x	x						x			x	c	x	x										x		x							x		
Spatial Product Analysis (Preparation)	Chapman and Pinfold, 2001			x				x					x								x		x	s	x							x				x		x							x		car structures
	Corallo et al., 2009			x				x					x								x		x	s	x							x				x	x										turbine components
	Bylund et al., 2004			x				x					x								x		x	s	x							x				x	x										jet engine & car struct.
	Johansson, 2014			x		x	x						x	x							x	x	x	s				x				x				x	x										ski rakcs
	Johansson, 2015			x		x	x						x	x							x	x	x	s				x				x				x		x							x		
[…] (Integration)	Vosgien et al., 2012; Vosgien, 2015			x		x	x	x	x					x								x	x	s	x								x			x	x										Integrated Power Plant System
[…] (Simulation)	Andrae and Koehler, 2016	x				x	x		x			x		x								x	x	s								x				x	x										pressure vessel

Table 5-1 continued

Design Automation Task (*derived*)	Reference	Input									Output								Goals										Design Automation Methods																		Case Study
																			Syn.		Analysis				Req. & Con.				Knowledge Representation										Reasoning Technique								
		function	prod.arch.	geometry	process	comp.lib.	intercon.	modific.	mapp.	eval.alg.	function	prod.arch.	geometry	simulation	process	BoM	cost est.	stan. model	topology	parameter	simplification	integration	simulation	Level	fun.con.	spa.con.	obj.fun.	des.var.	grpah	port-based	bond graph	obj.-or.	UML based	ontology	shape based	CAx	proc.	decl.	Combinatorial	search, det.	search, non-det.	constr.-sol.	ont.-based	interact.	inf.-eng.	grammar	
Spatial Topology Optimization	Patel and Campbell, 2010			x				x					x						x	x				c		x	x		x												x					x	sheet metal
[…] incl. Simulation	Shea et al., 1997; Shea et al., 2005; Shea and Smith, 2006			x			x	x					x						x	x			x	s	x	x	x								x						x					x	truss structure design; stadium roof structure; transmission tower
	Cui and Wang, 2013					x	x	x	x				x						x	x			x	s				x				x				x	x				x						ship cargo tank struct.
	Bolognini et al., 2007; Starling and Shea, 2005; Hooshmand and Campbell, 2014			x			x	x	x				x						x	x			x	s	x	x	x		x											x						x	micro-electromechanical systems; gear trains; fluid channel
Design Configuration	Van der Laan et al., 2008; Mandorli et al., 2002; Danjou et al., 2008; Schotborgh et al., 2012; Elgh, 2007; Raffaeli et al., 2013; Frank et al., 2014; Emberey et al., 2007; Bermell-Garcia et al., 2012; Frank and Hillbrand, 2012					x	x	x					x						x	x				s				x				x				x	x	x							x		composite sheet ribs and molds; press brake machine; compressor components; car seat heating; gas turbine exhaus; platforms; aircraft skin panels; aircraft composite materials; fastening mechanisms
	Colombo et al., 2005; Gerhard and Lutz, 2011; Di Gironimo, 2014					x	x	x					x			x			x	x				s				x				x				x	x	x							x		press brake machine; stamping tools; train traction control units
	Chavali et al., 2008					x	x	x					x						x	x				s				x				x				x	x	x							x		grid assembly
	Ruschitzka et al., 2010					x	x	x					x						x	x				s				x						x		x	x						x				trimming packets
	Sunnersjö et al., 2006					x	x	x					x			x			x	x				s				x				x				x	x										submarine bulkhead
[…] incl. Analysis of Components	Colombo et al., 2015; Choi, 2009					x	x		x				x	x					x	x	x		x	s				x				x				x	x										compressor comp.; aircraft wing struct.
Product Platform Design	Siddique and Rosen, 1999		x															x	x					s					x											x							automotive family design
	Tarkian et al., 2011			x		x	x	x		x								x		x			x	s	x	x						x				x	x				x						robot family design
Transformation Process Planning (Co-Evolution of Product and Processes)	Stanković et al., 2012				x	x	x								x				x					s					x										x							x	stiffened panel ass.
	Weilguny and Gerhard, 2009					x	x	x					x		x				x	x				s				x				x				x	x										bores of engine caseing
[…] (Design of Fixtures)	Gmeiner and Shea, 2013			x		x	x	x	x				x						x	x				s										x									x				jaw fixture design

Table 5-2: Definition of design automation task categories and corresponding information on in- and outputs, goals as well as characteristics of available design automation methods

Task Description	Description	Input									Output								Goals										Design Automation Methods																	
																			Syn.		Analysis				Req. & Con.				Knowledge Representation										Reasoning Technique							
		function	prod.arch.	geometry	process	comp.lib.	intercon.	modific.	mapp.	eval.alg.	function	prod.arch.	geometry	simulation	process	BoM	cost est.	stan. model	topology	parameter	simplification	integration	simulation	Level	fun.con.	spa.con.	obj.fun.	des.var.	grpah	port-based	bond graph	obj.-or.	UML based	ontology	shape based	CAx	proc.	decl.	Combinatorial	search-det.	search-non-det.	constr.-sol.	ont.-based	interact.	inf.-eng.	grammar
(1) Functional Synthesis	Derivation of functional structures by decomposition of a main function or existing functional models with consideration of model libraries	x				x	x				x								x					?	x				x					x						x	x		x			x
(2) Product Architecture Topology Synthesis	Integration of functional components based on system boundary conditions and functional constraints, e.g. load case scenarios. No geometrical aspects considered.	x				x	x	x	x			x							x					s	x				x	x	x	x		x			x		x	x	x	x	x			x
(3) Product Architecture Parameter Synthesis	Variables of functional components are determine for a given product architecture topology.		x					x		x		x								x				s	x		x				x	x										x		x		
(4) Product Architecture Analysis	Functional performance is assessed based on heuristics or mapping to dynamic simulation models.		x				x	x	x					x									x	s	x		x					x				x	x									
(5) Spatial Product Architecture Topology Synthesis	Geometric instances are assigned to functional components. Topology can be modified by exchange, addition or removal of components as well as change of parameters.			x		x	x	x		(x)			x	(x)					x					s	x	x	x	x					x		x		x				x			x		x
(6) Spatial Product Architecture Parameter Synthesis	For a fixed topology, parameters are modified in order meet functional and spatial constraints.			x				x		x			x							x				s	x	x						x	x			x	x			x	x	x			x	
(7) Spatial Component Architecture Synthesis	On the component level, parameters and features of shapes are modified to meet functional and spatial constraints.			x				x					x						x	x				c	x	x						x			x	x	x			x				x		x
(8) Spatial Product Analysis (Preparation)	Geometric models are transformed (simplified, meshed, segmented) to yield simulation ready models.			x				x	x				x								x			s								x				x		x				x			x	
(9) Spatial Product Analysis (Integration)	Multiple simulation models are integrated to one overall model so that system performance can be analyzed.			x		x	x	x	x					x								x		s									x			x	x									
(10) Spatial Product Analysis (Simulation)	Loads and boundary conditions are applied and simulation parameters determined so that Simulations can be executed automatically.			x				x						x									x	s	x	x						x				x	x									
(11) Spatial Topology Optimization	Fully integrated loop of spatial product architecture synthesis (top.&par.) and analysis guided by optimization algorithms			x		x	x	x	x					x					x	x	x	x	x	s	x	x	x		x			x			x		x				x					x
(12) Design Configuration	Integration of (parameterized) components based on a predefined tree of possible configurations according to characteristic design variables.					x	x	x					x		x	x	x		x	x				s				x				x		x		x	x	x				x	x		x	
(13) Transformation Process Design (co-evolution of design)	Co-evolution of products and corresponding processes during the early stages of design to assess mutual impacts				x	x	x								x				x					s	x	x			x			x				x	x		x	x						x
(14) Transformation Process Design (production plan)	Production plans are created based on (geometrical) product characteristics.			x		x	x	x							x				x	x				s								x				x	x	x		x					x	
(15) Transformation Process Design (attributes)	Process attributes required for manufacturing are designed.			x		x	x	x							x				x	x				s		x								x									x			
(16) Product Platform Design	Derivation of standardized modules in order to establish a product platform, or reduce its size.			x		x	x	x		x								x	x	x				s	x	x	x		x			x				x	x			x	x					

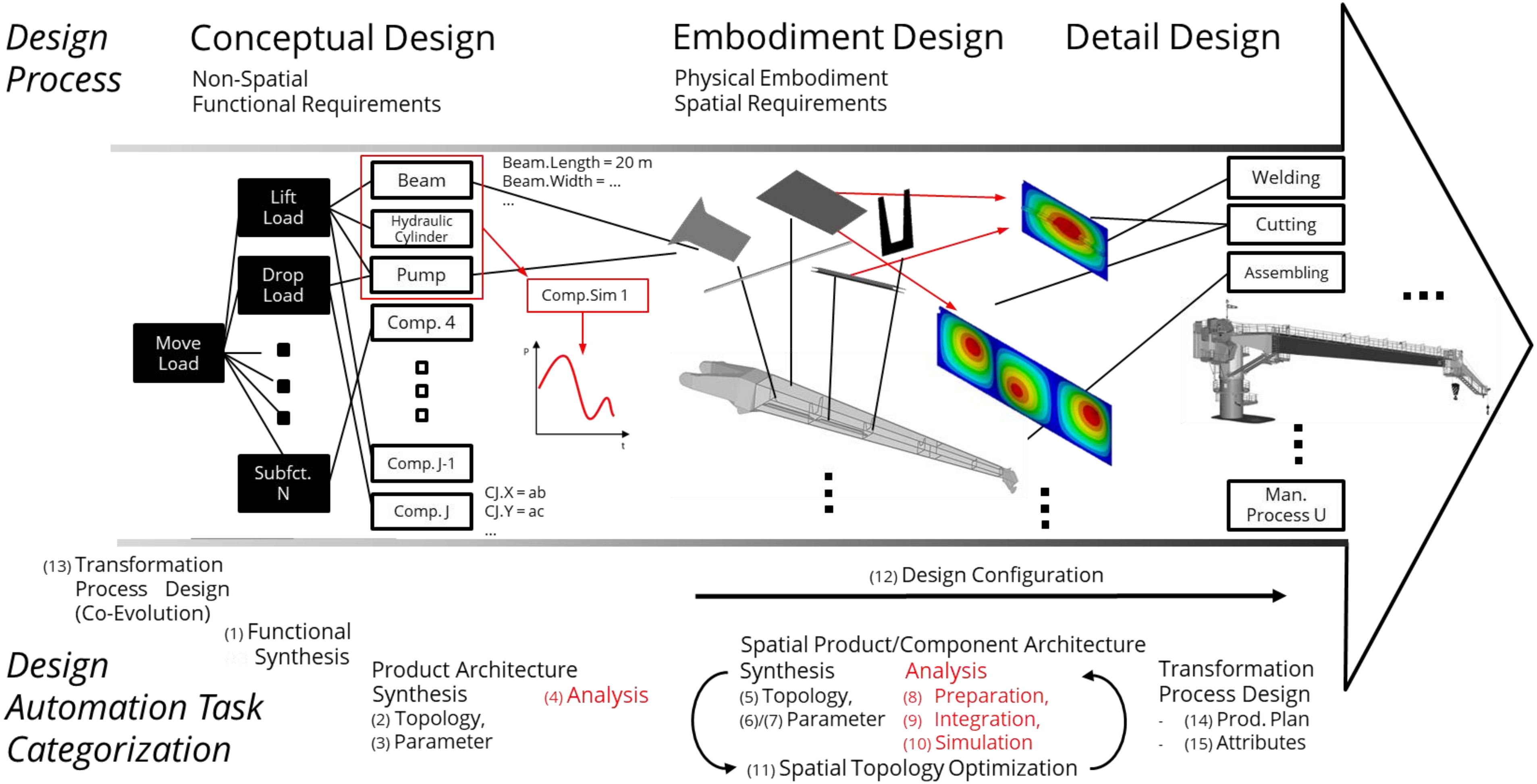


Figure 5-3: Overview of design automation task categories and relations with regards to the design process in an application context of crane system development. The numbering of categories refers to the numbering introduced in Table 5-2.

5.3 Derivation of Design Automation Task Templates

To provide the basis for the methodology for design automation task definition, design automation task templates are required. As indicated in Section 2, existing templates need to be enriched to account for the specifics of state-of-the-art design automation opportunities. The templates need to account for the knowledge levels required for design automation task formalization which are unique for each design automation task category. In particular, the templates detail the knowledge of inputs, outputs and goals. To support the different aspects of the methodology for design automation task definition, templates to support design process integration, estimation of the impact of design automation on design practice and product knowledge formalization are required. Thus, templates that take into account the process perspective, the motivational perspective and the product knowledge perspective need to be distinguished. In alignment with the meta-model presented in Figure 4-2, the templates build upon the ArchiMate and SysML languages and are illustrated based on the knowledge that characterizes a generic design automation task.

5.3.1 Design Automation Task Templates – Process Perspective

Considering the integration of a design automation task to a design process model, Figure 5-4 depicts how a generic design automation task template can be represented in the ArchiMate language. The generic template serves as a basis for derivation of design automation task templates for each type of design automation task categories so to account for the specific types of knowledge needed. The design automation task template capture the relevant knowledge of the related design automation activity as well as the specifics of the supporting design automation application. The design automation activity belongs to a design process and automatically transforms input data to output data based on the design automation application and according to the goals formalized by the designer. The design automation application that is shown at the bottom of Figure 5-4 accounts for the design automation methods that are available for a specific type of design automation task. With respect to design automation integration to the design process, the knowledge related to input, output and goals is either available explicitly in the form of data depicting product models or captured tacitly as expert knowledge. Thus, potential design automation use cases can be identified and validated based on availability of formalized knowledge.

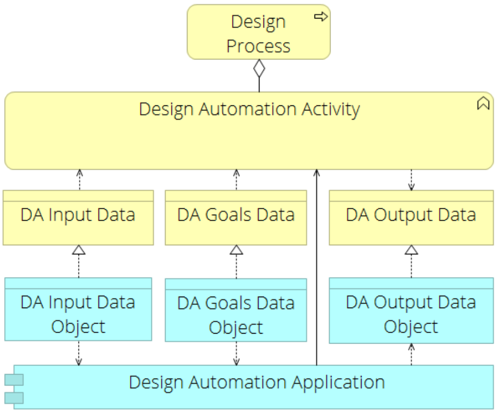


Figure 5-4: Generic design automation task template – process perspective

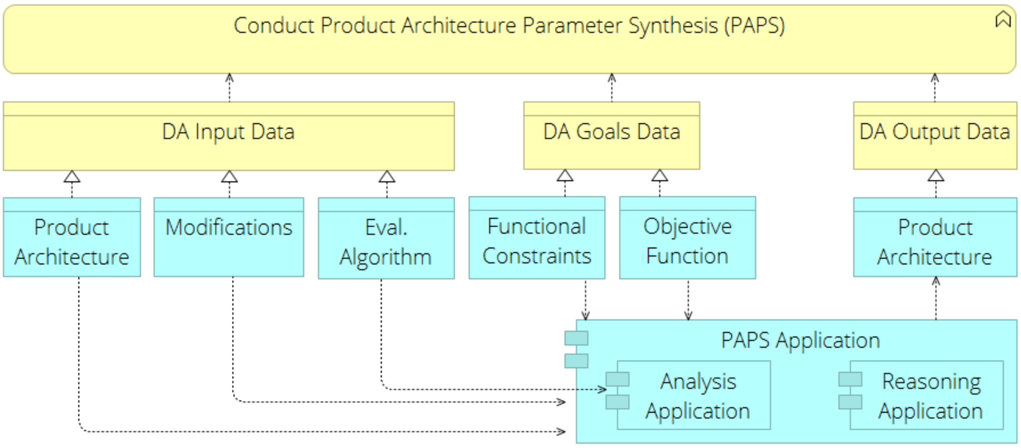


Figure 5-5: Design automation task template for product architecture parameter synthesis

Figure 5-5 shows the design automation task template for the category Product Architecture Parameter Synthesis (PAPS). In alignment with the design automation task categorization, data objects are specified in more detail with respect to input, output and goals. Considering the input product knowledge, data on the product architecture, modifications and evaluation algorithms for assessment of functional performance of designs are needed. For the case of PAPS, modifications refer to parameters and variables. Regarding the evaluation algorithms, the PAPS Application aggregates an analysis application so to execute the evaluation algorithm or run an external simulation. The knowledge with respect to goals needs to be captured regarding the functional constraints as well as the objective function so to serve as input for the reasoning application. Lastly, the output corresponds to the product architecture that is updated with respect to its parameters.

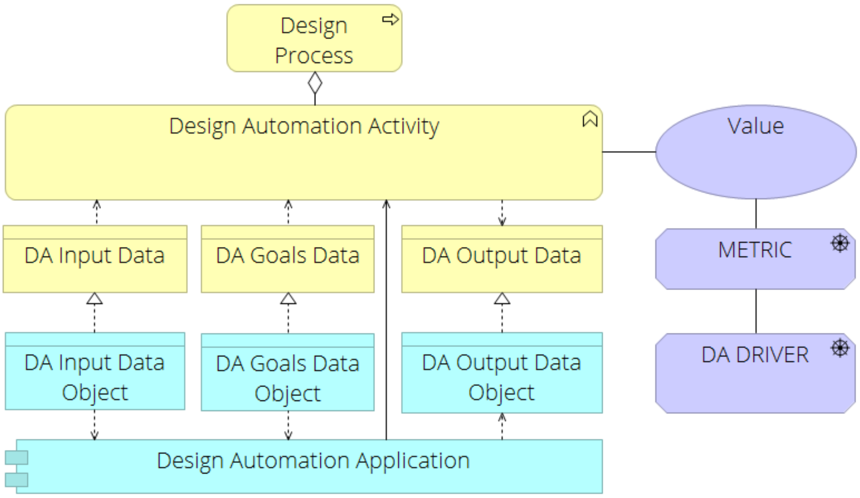


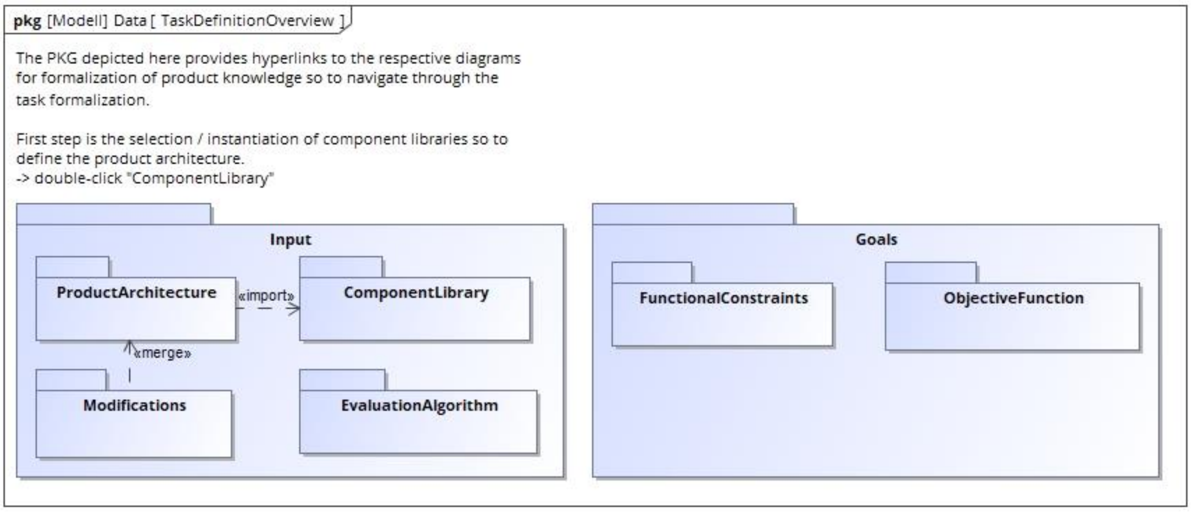
Figure 5-6: Generic design automation task template – motivational perspective

5.3.2 Design Automation Task Templates – Motivational Perspective

The literature review presented in (Rigger, Münzer, and Shea 2016) showed that specific metrics can be mapped to specific design automation task categories. Figure 5-6 illustrates how the value of design automation can be mapped to a design automation activity. The value itself is quantified based on one or more metrics and related design automation drivers whereby the latter serve as a means to categorize the metrics that are relevant for a specific design automation task template. Thus, the motivational aspect of design automation can be integrated to the design automation task templates.

5.3.3 Design Automation Task Templates – Product Knowledge Perspective

As indicated in Figure 4-2, the SysML BDD and PAR diagrams can be associated to data objects identified in the design process. For organization of the formalized product knowledge and enabling modularization of knowledge, SysML package diagrams (PKG) are defined. Hyperlinks attached to packages permit navigation through the model. Further, a description of the steps to be conducted for modeling the relevant product knowledge is depicted on each diagram. Figure 5-7 shows the first view of the SysML template for a PAPS task. It can be seen that the packages contain the same elements as the corresponding process template, see Figure 5-5. Whereas the package “ComponentLibrary” is imported by the package “ProductArchitecture” to indicate the reuse of elements from model libraries, the package “Modifications” is merged with “ProductArchitecture” since Parameters/Variables that are the relevant modification for PAPS can be directly indicated in the “ProductArchitecture”. The specifics of the formalization are detailed in Section 9. In this section, the aim is to demonstrate the template-based approach for guiding knowledge formalization for design automation task definition.



The PKG depicted here provides hyperlinks to the respective diagrams for formalization of product knowledge so to navigate through the task formalization.

First step is the selection / instantiation of component libraries so to define the product architecture.
-> double-click "ComponentLibrary"

Figure 5-7: Design automation task template for formalization of product knowledge for PAPS

5.4 Discussion

Based on the characteristics of design automation methods reported in scientific literature, a design automation task categorization is derived that puts into context the knowledge levels required for implementation of design automation, design automation methods and the design process. Further, design automation task templates are derived based on the task categorization to support for different perspectives of design automation task definition. In particular, templates are derived that take into account the design process perspective, the motivational perspective and the product knowledge perspective to support design process integration, estimation of the impact of design automation on design practice and knowledge formalization, respectively. In the following, the characteristics of the knowledge levels required for implementation of design automation in design practice are discussed. Following this, the differences and similarities of the research field KBE and CDS are analyzed and potential for consolidation is indicated. In particular, the characteristics of design automation tasks addressed by methods from KBE or CDS are investigated as well as the attitude of design automation methods with respect to potential benefits of implementation. Following this, knowledge formalization and reasoning techniques are analyzed before the discussion closes with the analysis of the maturity of design automation methods presented in literature.

5.4.1 Knowledge Levels for Design Automation Task Definition

Analyzing the *required input knowledge* for design automation tasks, it can be seen that independent from the design state, design automation tasks related to synthesis of architectures (functional, product …) require the definition and formalization of component libraries and knowledge of interrelations of components, i.e. well established building blocks. Therefore, this requires modularization/standardization of the relevant design knowledge according to company specific modularization/standardization principles (Andreasen 2011). This partially confirms the statement of Schut, Kosman, and Curran (2013) that standardization is required for implementation of design automation. Further, the categorization indicates that additional product knowledge with regards to performance assessment of generated designs is required in order to evaluate new generations of designs for the synthesis of products. Consequently, fundamental knowledge on the design domain is required in order to enable algorithmic evaluation of designs based on heuristics or generation of dynamic simulation models. The key challenge within mechanical system design is to handle the many-to-many mappings of functional building blocks to its physical realization. Resolving this problem has been tackled by recent design synthesis methods by means of incorporating complex simulation techniques for assessment of generated designs, e.g. Münzer and Shea (2015) and dynamic generation of simulation models for changing geometries (Zimmermann, Chen, and Shea 2018). However, industrial application of the approaches is pending. For design automation tasks focusing on topology synthesis, the formalization of the system boundary accounts for the formalization of modifications: what are the degrees of freedoms, which interfaces need to be considered etc. In contrary, for design automation tasks addressing parameter synthesis, modifications refer to the indication of parameters and variables that are subject to change. Thus, the categorization loses expressiveness at the cost of generality. Potentially, a refinement of the characteristics supports understandability of the design automation task categorization to practitioners.

Analyzing the *required knowledge on goals*, the distinction of addressed system levels is not feasible for the case of functional synthesis since functional models are synthesized independent from the physical objects that are used for realization of the design (Pahl et al. 2007; Ulrich and Eppinger 2004). On the other hand, functional models themselves can be used as starting point for synthesis of product architectures (Helms and Shea 2012; Campbell, Cagan, and Kotovsky 1999). However, the functional modeling is rarely applied in an industrial context. Correspondingly, the design automation applications addressing functional modeling remain mainly bound to academia. Thus, more recent work focuses on definition of system boundaries that contain the specification of functional interfaces for product architecture topology synthesis, e.g. through ports indicating energy flows (Münzer and Shea 2017; Kerzhner 2012). In order to further reduce the level of abstraction, design automation aiming at design configuration requires definition of functional performance values that enable a one-to-one mapping of functional requirements to physical components. Hence, characteristic design variables guide the design configuration of the product. This requires deep understanding of the product, so to be able to derive these parameters. If this is not possible, objective functions or functional and spatial constraints can be defined to characterize the solution space and constrain and guide the solution search.

Lastly, to further extend the scope of the categorization, additional literature from other fields such as systems engineering, design by analogy and methods and tools from software suppliers need to be investigated. This potentially enriches the categorization by means of additional categories and also criteria for analysis. Regarding the validation of the proposed categorization, integration of design automation task templates that reflect the knowledge levels to industrial processes is needed. In this respect, Section 6 that presents the method for identification of design automation use cases based on templates presents the validation of different design automation task templates.

5.4.2 Consolidation of Research Fields

The categorization of *design automation tasks* lists 16 distinct task categories ranging from automation of technical process synthesis over functional synthesis to automation of embodiment and detail design. The majority of methods stemming from the research field of KBE focus on design automation tasks involving geometric representation, parametric modeling and advanced CAD applications, see Table 5-1. This implies the development of CAD based product configurator support systems, which in many cases present a frequent topic of interest in KBE research. The practical orientation of KBE is further strongly established with links to manufacturing process support. Summarizing, KBE predominantly aims at automation of routine design tasks and configuration of modularized products. In contrast, CDS emerged from a unique research direction with interest to computationally support the early stages of the design process. It can be observed that methods aim at solution space exploration and generation of variants at a stage of the design process where the degrees of freedom for design are high. Based on Table 5-1 CDS is described in terms of model-based engineering established through functional modeling and simulation-based design, which is a frequent approach utilized to model the structure and behavior of technical products at the conceptual and early embodiment design stage. This altogether results in a division between the two research fields with KBE automating embodiment and detail design tasks, and CDS aiming at conceptual design and early embodiment design. Hence, both KBE and CDS address embodiment design tasks such as spatial product architecture parameter synthesis.

However, the *attitude of the available design automation method* determines what method to use in a given context. Whereas KBE methods strictly constrain the solution space by application of rules for derivation of outputs based on formalized goals, CDS methods apply general search strategies for solution space exploration. Even though the different opportunities are captured within the available knowledge formalizations and reasoning methods per category, the information what method to use in what context is currently not captured in the categorization. Yet, the templates addressing the motivational aspects of design automation per category can be extended to also capture the specifics of each method, e.g. as shown in (Rigger, Münzer, and Shea 2016).

Regarding *knowledge formalization*, it can be seen in Table 5-2 that a variety of representation techniques are used design automation. This hinders exchange and reuse of knowledge among systems. Whereas for the case of CDS methods predominantly graph based formalizations are applied, rule-based systems incorporating object-oriented programming paradigms are mostly applied for the formalization of knowledge for KBE systems. As indicated in Section 2.3, SysML appears to be a promising candidate for consolidating efforts regarding design automation task formalization for design automation methods stemming from both communities. Efforts towards usage of SysML exist for both communities, e.g. (Kerzhner 2012; Shah et al. 2012) for CDS and (Klein, Lützenberger, and Thoben 2015) for KBE. The SysML task templates as introduced in 5.3.3 build upon both the task structure as well as the possibilities of SysML modeling environments to define packages and different views to organize the knowledge in models. Hence, they provide a convenient means to organize, assess and navigate within models. The issue of doubling knowledge, for instance by storing product knowledge of a CAD model to a SysML model might be mitigated by the advance of systems engineering techniques in industrial environments and the corresponding formalization of product models as well as efforts towards explicit formalization of knowledge.

With respect to *applied reasoning methods* for design automation, a plurality of possibilities can be found for a given design automation task. As shown in Table 5-2, for automation of routine design activities as addressed by KBE methods, procedural reasoning methods are available for almost all categories. The sequence of rules application is predetermined by procedural rules that are encoded directly or derived using more information rich concepts such as Design Structure Matrices (DSMs) (Eppinger and Browning 2012). In contrary, CDS methods most often pursue general search strategies such as depth-first search or other types of search-based optimization algorithms to reason on the provided input product knowledge. For grammar based systems, the design space is altered based on a predefined rule set. The selection of this set of rules substantially impacts the solutions space and is per se a non-trivial task (Königseder, Corinna 2015). If used in combination with heuristic search techniques, the stochastic nature of the algorithms allows the support system to both explore and exploit the design space for generation of possibly creative solutions to the design task. In contrary, the usage of declarative formalizations is applied by KBE methods for configuration tasks, however, have also been applied to conceptual design automation tasks such as product architecture topology synthesis (Münzer and Shea 2015). In order to enable a declarative formalization of the design automation task, the task needs to be of the type of a general configuration problem (Zhang 2014) including well-defined component interfaces and dependencies. Regarding the use of ontologies for knowledge representation and reasoning, only few works have been published even though they allow the handling of diverse and complex design knowledge and corresponding relationships occurring during the design process. Appropriate standards for encoding ontologies are an open issue (Chandrasegaran et al. 2013).

With respect to the *maturity of methods*, it has to be noted that most methods stemming from the KBE community are applied in industrial contexts and commercial platforms for implementation of solutions exist. On the other hand, the plurality of methods from the CDS community have not been applied to industrial test cases, however, are developed for academic demonstration of concepts and methods and are applied to academic test cases imitating industrial settings. To foster transition of design automation methods from academia to industry, the value of application needs to be communicated and designers need to get familiarized with methods. Multiple approaches can be taken: methods are needed that are able to comprehensively communicate the opportunities and value of design automation methods given a design process in industry. Second, engineering curricula should account for design automation implementation to familiarize novel designers with available methods and the underlying concepts. Third, expert designers need to get acquainted with the concept of design automation. Hence, small-scale design tasks should be addressed first, before automation of more complex design tasks based on more advanced design automation methods.

5.5 Summary

This section presents the derivation of a design automation task categorization and related templates based on analysis of 77 application oriented methods from the research fields KBE and CDS. Thereby, means to increase awareness of opportunities of design automation in industry are provided based on knowledge levels required for design automation task definition. Further, design automation task templates, textual description of categories as well as a graphical overview that relates the design automation tasks to a generic design process support identification of design automation use cases in design practice. The categorization is derived based on the analysis of characteristic criteria of design automation tasks. In particular, input knowledge, goals and output knowledge are investigated. It is shown, that the addressed output and purpose of different design automation methods enable to build 16 clusters that represent the distinct design automation task categories that account for design automation tasks ranging from support of technical process synthesis to detail design tasks. Thus, the properties of the design automation task categories provide an answer to research question one: "*What are the characteristics of design automation tasks that enable identification of design automation opportunities in design* *practice?*" In relation to the first research question, the **first contribution** can be highlighted as follows:

The establishment of a design automation task categorization and its characterization of tasks based on knowledge levels with respect to input, goals and output.

Further, it is shown that the knowledge levels needed for design automation task definition enable derivation of design automation task templates based on task characteristics. These templates take into account the design process perspective, the motivational perspective and the product knowledge perspective. The templates are used in the remainder of this work to support the methods presented in Sections 6 - 9. Additionally, a consolidation of the research fields KBE and CDS is conducted to provide the conceptual basis for development of the methods for design automation task definition. This allows taking into account the specifics of each field, using synergies and further streamlining design automation task definition towards the goals of design automation method independent task definition. In particular, the knowledge formalization, reasoning methods, maturity and scope of investigated design automation methods are assessed. Hence, the **second contribution** can be summarized as follows:

The consolidation of the research fields KBE and CDS based on systematic analysis of characteristics of design automation methods presented in literature.

The following sections gradually introduce the methods that build on the results of derivation of the design automation task categorization.

6 Method for Identification of Design Automation Use Cases

The first step of the methodology for design automation task definition accounts for the identification of design automation use cases in design practice. In the past, research efforts have focused on the development of design automation methods to computationally support design. Yet, supporting methods that focus on the identification of design automation use cases and corresponding integration of design automation in design practice are missing. In particular, support for decomposition and analysis of design processes and supporting technological environments are needed to enable comprehensive integration of design automation.

In the following, Section 6.1 proposes a method that is based on a five step procedure to support the identification and integration of design automation in industry. After introducing the basic concepts of the proposed method, the steps are presented in detail. In Section 6.2, the results obtained from application of the proposed method to three industrial cases are presented. Next, Section 6.3 critically assesses the results yielded from evaluation and finally, Section 6.4 presents a summary of the section highlighting the contributions of the method to answer research question two as well as its implications to design research and practice.

6.1 Method

The proposed method for identification of design automation use cases builds on the design automation task categorization presented in Section 5.2. In particular, the design automation task templates that account for the design process perspective are applied for identification of knowledge required for design automation task definition, see Section 5.3.1. A two-stage analysis of the design process constitutes the core of the method. First, the meso-level of the design process is modeled so as to capture end-to-end task flows (Wynn and Clarkson 2018). Second, the micro level of the design process is modeled in order to account for the individual process steps and activities (Wynn and Clarkson 2018). In this respect, focus is put on integrated modeling of the design process with workflows, applications of design process support methods and corresponding tools and technologies. This enables mapping of data and corresponding tools and technologies to design automation task templates for identification and validation of design automation use cases. Based thereon, potential future scenarios of design processes that take into account design automation can be modeled. Hence, the integration into design practice including the technological environment can systematically be taken into account and acknowledged already for identification of use cases. To capture the different viewpoints of practitioners on the design process according to their position and experience, collaborative workshops are carried out. The yielded information on the design process is modeled live on screen during the workshop using the ArchiMate 3.0 modeling language and the open-source modeling software Archi 4.0.2. Thus, the participants can directly validate the information captured in the model.

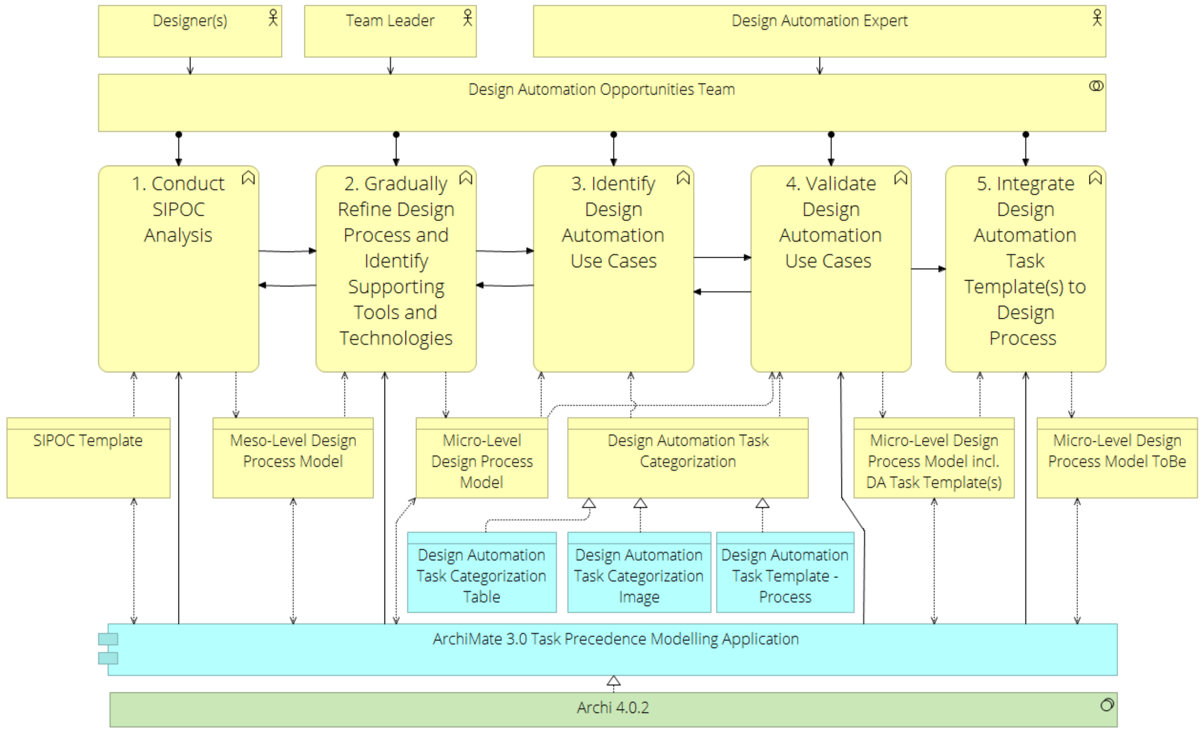


Figure 6-1: Method for identification of design automation use cases. The figure highlights the method's main steps including respective data flows (central layer, yellow), the workshop composition (top layer, yellow) as well as supporting tools and technologies (blue & green).

The method can be summarized in five steps that account for meso- and micro-level modeling of the design process, search for design automation use cases in the design process model based on the design automation task categorization, validation of the use case based on implementation of the design automation task templates and modeling of potential future scenarios of the design process. In the following, first the five step procedure outlined in Figure 6-1 is detailed and an illustrative example is provided before the workshop setup is presented:

1. A Supplier-Input-Process-Output-Customer (SIPOC) analysis (Yang and El-Haik 2003) is conducted to frame the scope of the investigated design process on a meso-level. In the context of design process, the SIPOC analysis is used for definition of the main tasks (P) that are described by its input data (I) and output data (O). Further, the corresponding suppliers (S) and recipients (C) of information are indicated. For the workshop, an empty template of the SIPOC is projected on a wall and gradually defined with the participants through questions regarding the five pillars of the SIPOC: who are the suppliers, what data do they

deliver, what are the main tasks, what's the corresponding output data that is generated and who is it for. Figure 6-2 shows the empty SIPOC template including a generic example of a design process with three tasks.

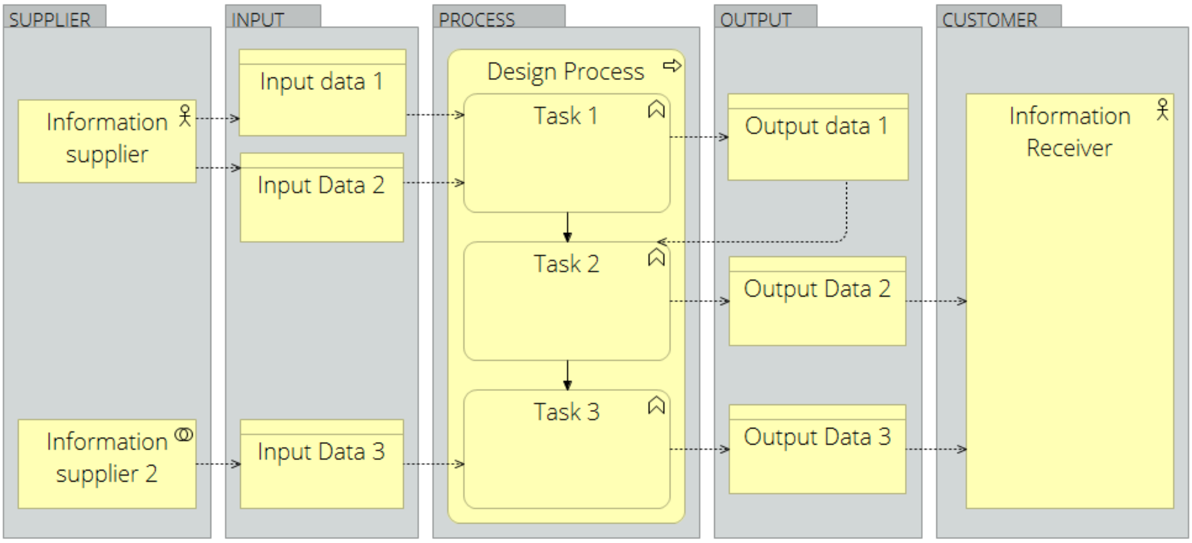


Figure 6-2: Template for SIPOC used as a starting point for reengineering of the investigated part of the design process generated in Archi Software. The example depicts three generic design tasks and respective in- and outputs.

2. The meso-level process attained by the SIPOC analysis is gradually refined with respect to the design process sub-activities so as to yield a micro-level process model (Wynn and Clarkson 2018). Further, related tools and technologies are mapped to the design activities. In relation to design tasks, design activities are assigned resources as well as goals that specify the outputs (Duffy 2005). Yet, the activity goals are depicted on diagrams only if necessary. In the workshop, questions are posed related to tools and technologies that are used to support an activity, how the tools work and what sort of implementations exists. The refinement of the process is performed until the level of granularity where tools and technologies can be mapped to individual design activities. The yielded information is modeled live on screen so that participants can directly validate the information captured in the model. Figure 6-3 continues the example introduced in Figure 6-2 and depicts the sub-activities and supporting software applications of Task 1. In particular, the generation and evaluation of a design layout is depicted for which – in case the evaluation is successful – the documentation is generated. To simplify, the implementation layer (green) is omitted in the example shown in Figure 6-3.

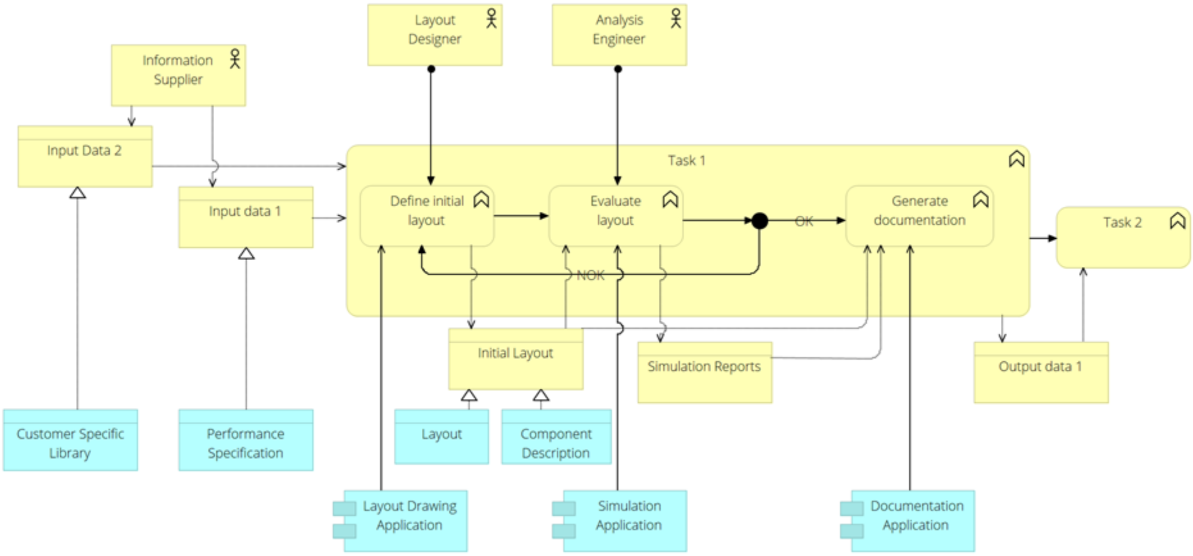


Figure 6-3: Example of a micro-level process model continuing the example introduced in Figure 6-2.

3. Both the graphical contextualization of the design automation task categorization with the design process according to (Pahl et al. 2007) (Figure 5-3) and the tabular listing of design automation task categories (Table 5-2) with detailed information of knowledge needed for design automation task definition are applied for identification of appropriate design automation task templates. Thereby, the visualization supports identification of design automation task categories fitting the investigated design process with respect to the design stage, e.g. conceptual design or embodiment design. Assuming that “define initial layout” in the example from Figure 6-3 refers to the definition of topologies of concepts, e.g. for hydraulic systems, the design automation task category “product architecture topology synthesis” that belongs to conceptual design can be identified as a design automation use case: “Integration of functional components based on system boundary conditions and functional constraints, e.g. load case scenarios. No geometrical aspects considered.” (cf. Table 5-2).

4. The templates identified in Step 3 are used to map the relevant data objects identified within the design process model gained in Step 2. Potentially, distributed sources of the data object exist. For example, data on component libraries can be spread among multiple databases and documents containing design guidelines. The allocation of data objects to specific formalization enables identification of what type of data needs to be gathered, aligned and formalized to account for the knowledge required for design automation. Particular attention has to be paid for the formalization of goals due to often recognized ill-defined problem formulation with vague or incomplete goals in design (Dinar, Danielescu, et al. 2015). Based on the mapping of inputs, outputs and goals, a validation of the design automation use cases can be conducted. In particular, the availability of (formalized) knowledge for design automation task

formalization is assessed. For the present example, the instantiation of the template related to “product architecture topology synthesis” is shown in Figure 6-4. It can be seen that currently, the knowledge for evaluation is not formalized since being captured tacitly by analysis engineers. Correspondingly, there are no mapping rules to simulation components available.

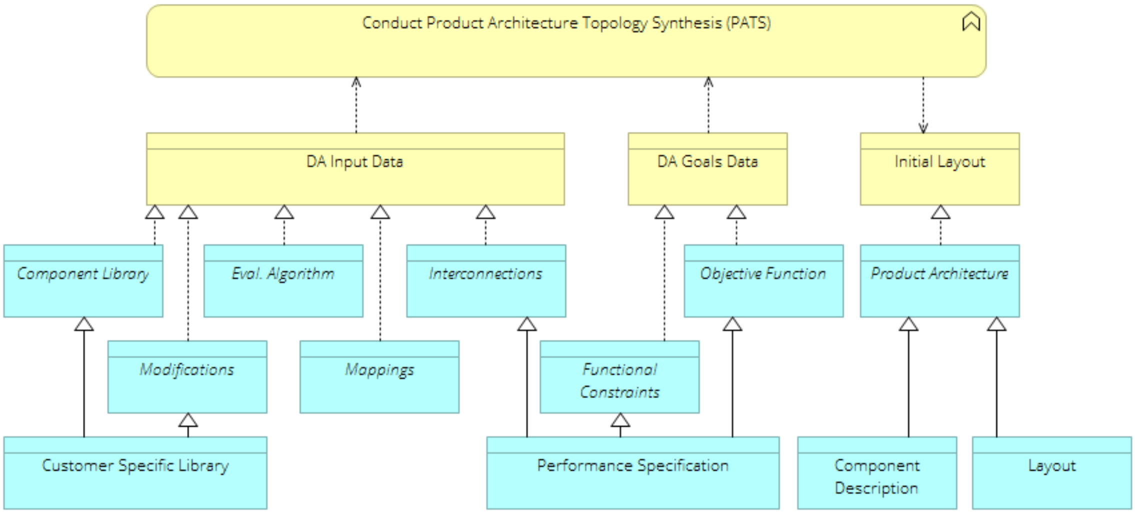


Figure 6-4: Instantiated PATS Template for the example introduce in Figure 6-2.

5. Once the different design automation use cases are identified, the potential future scenarios of the design process can be modeled by means of integration of the instantiated design automation task templates to the design process model. The obtained variants of the design process are used for highlighting one or multiple opportunities for design automation implementation in practice as well as the impacts on the structure of the design process so that design practitioners attain an impression of how design automation alters design practice. Figure 6-5 shows the updated model of the design process for the present example with the PATS template integrated. It can be seen that the integration puts several requirements on the existing process, e.g. the information supplier needs to provide the performance specification in a format interpretable by the design automation application. Further, the design automation application needs an interface to the existing simulation application and provide capabilities for generation of reports in order to avoid additional (manual) activities. On the other hand, the analysis engineer does not need to conduct any manual activities for this process but participate in a review meeting to evaluate the results of the design automation application.

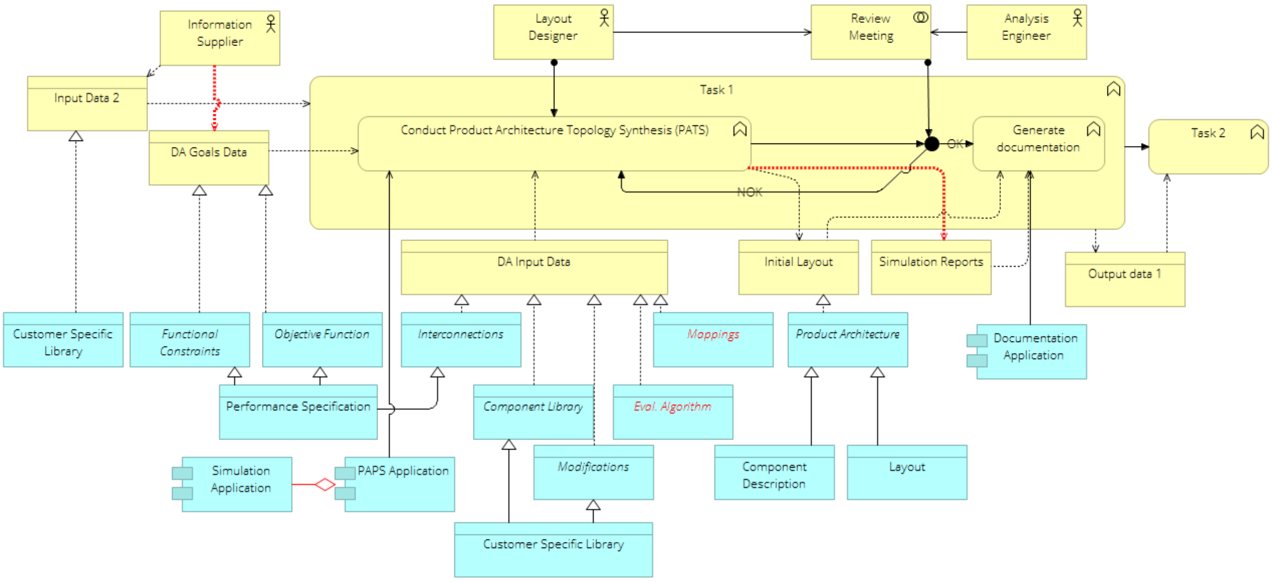


Figure 6-5: Example of the updated micro-level process model with integrated design automation task template. Elements in red highlight requirements for design automation implementation.

Workshop Organization

Regarding the workshop setup, at least one member of the workshop team should be familiar with the design automation opportunities as highlighted in the design automation task categorization. Further, one member needs to guide the workshop, introduce the necessary methods and tools of the method and has proficient design process modeling skills in the ArchiMate language to enable live modeling of the captured information. Preferably, these two workshop participants are not related in any means to the department in charge of the investigated design process. Thus, bias in analysis can be avoided. Regarding the selection of designers that are related to the design process, it has to be noted that at least two designers of the considered design processes as well as the team leader are required so as to attain both the high-level perspective of the team leader as well as the detailed insights of the designers. Thus, a shared understanding of the design process is yielded (Eckert and Stacey 2010). As indicated in Figure 6-1, Steps 1 and 2 are conducted within the collaborative workshop and Steps 3 to 5 are conducted by the design automation expert based on the information yielded in the workshop. However, the yielded results from steps 3 to 5 need to be successively discussed and validated in workshops with practitioners. The design automation expert needs to be familiar with the design automation task categorization and its characteristics.

6.2 Results

Application evaluation of the method for identification of design automation use cases was conducted for the three industrial cases introduced in Section 4.3. In the following, the workshop setup for the use cases is listed as well as the yielded results.

6.2.1 Workshop Setup

A collaborative workshop was conducted for each of the industrial cases introduced in Section 4.3. For each workshop, the author of this thesis acted as the design automation expert and presented and introduced the method and supporting concepts. Additionally, a working colleague contributed to the workshops by capturing the information on design processes obtained using ArchiMate models as described in Section 6.1. A projector was used to enable the workshop participants to collaborate in the gradual evolution of the design process models. The author and the working colleague who participated in the workshops are external to the organizations. Table 6-1 lists the workshop participants including the respective position in the department as well as work experience.

Table 6-1: overview of participants according to the industrial use cases

Participant	Position	Work experience
CASE 1 – Structural Analysis		
Designer 1-A	Designer, Team leader	4
Designer 1-B	Design Engineer	10
Designer 1-C	Department Manager	12
CASE 2 – Hydraulic Systems		
Designer 2-A	Designer, Team leader	11
Designer 2-B	Designer, Team leader	11
Designer 2-C	Department head	35
CASE 3 – Hydraulic Units		
Designer 3-A	Designer	10
Designer 3-B	Sales back office	15
Designer 3-C	Designer, deputy department head	15
Designer 4-C	Department head	15

6.2.2 Workshop Results

In the following, the results of the application of the method for identification of design automation use cases are summarized. First, the results of the meso-level analysis as conducted in Step 1 are presented. The corresponding SIPOC process diagrams for Cases 1 and Case 2 are depicted in Figure 6-6 to Figure 6-8. Following this, the micro-level analysis of design processes is conducted to yield refined models of the design processes including the supporting tools and technologies. In Figure 6-9, the micro-level process model for Case 1 – Structural Analysis is presented. The yielded models for Case 2 and 3 can be found in 11Appendix A.1. Table 6-2 lists the number of sub-activities as identified within Step 2 to summarize the characteristics of the results yielded for each case. Further, Table 6-2 lists the efforts for the workshops needed to conduct Steps 1 and 2 as well as the number of identified design automation use cases. The details on the identified use cases (Step 3) and the corresponding validation of design automation task templates (Step 4) are provided in Table 6-3 to 5-5. This includes the listing of identified interfaces to third party software for each design automation use case. Figure 6-10 depicts the instantiated task template for a product architecture parameter synthesis task for Case 2 – Hydraulic systems. It highlights that all knowledge is available but multiple formalizations for the same information exist. The remaining task templates for the identified design automation use cases can be found in 11Appendix A.2. 11Appendix A.3 shows the potential future scenarios of the design process with integrated design automation task templates for the cases 1 and 2 to further highlight the interfaces of design automation applications when integrated to design practice.

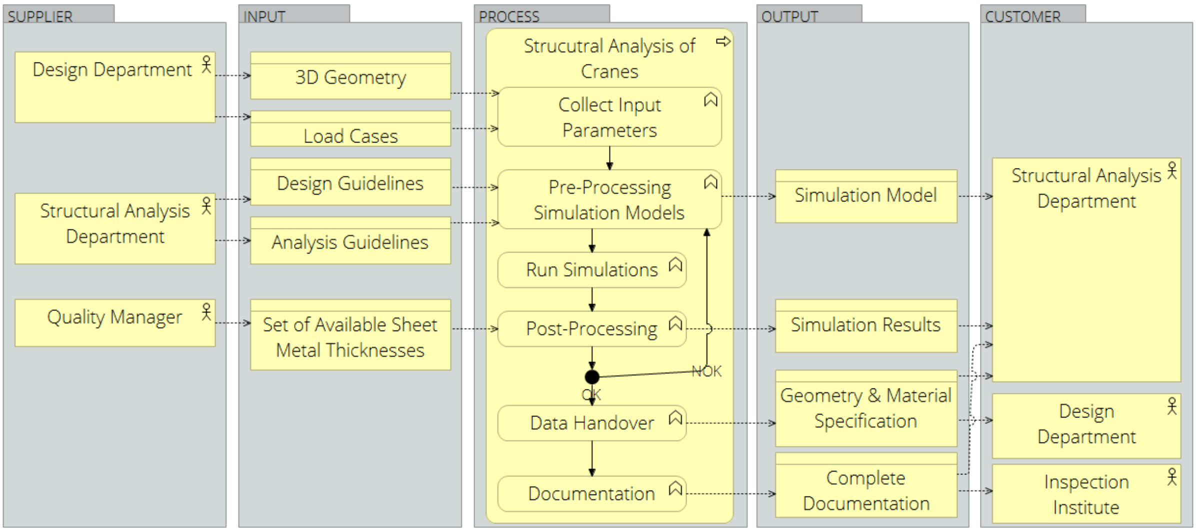


Figure 6-6: SIPOC yielded for Case 1 – Structural Analysis of Cranes

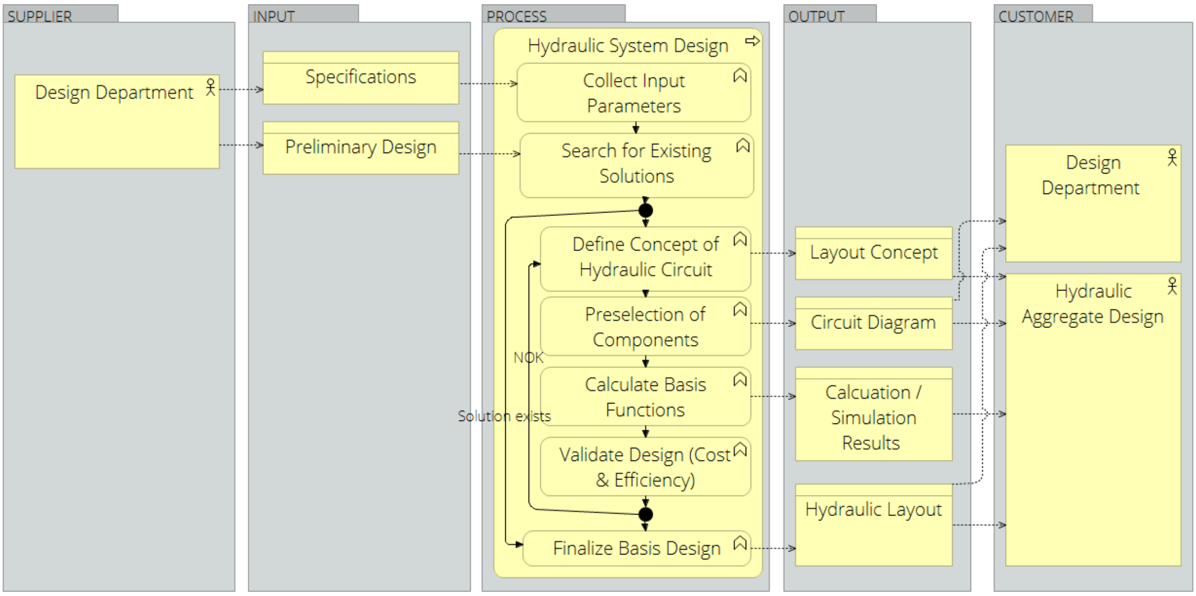


Figure 6-7: SIPOC yielded for Case 2 – Hydraulic System Design for Heavy Construction Machinery

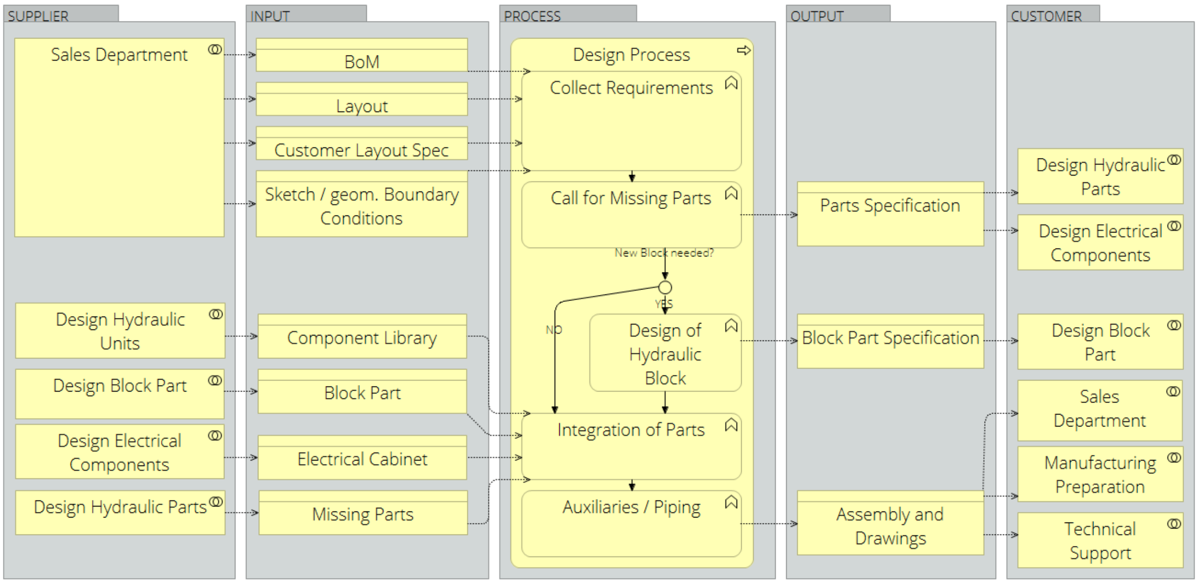


Figure 6-8: SIPOC yielded for Case 3 –Design of Hydraulic Units

Table 6-2: Results of application of the method for identification of design automation opportunities

Key Characteristic	Case 1 – Structural Analysis	Case 2 – Hydraulic Systems	Case 3 – Hydraulic Units
Number of additional activities identified in Step 2	17	25	34
Absolute Time for SIPOC analysis	1 h	½ h	¾ h
Absolute Time for Step 2	2 h	1 ½ h	1 ¾ h
Number of identified design automation use cases	2 (See Table 6-3)	2 (See Table 6-4)	3 (See Table 6-5)

Table 6-3: Characteristics of identified design automation opportunities for Case 1 – Structural Analysis

DA Task Template	Characteristic	Details
Spatial Product Architecture Parameter Synthesis	Use case	Automated sizing of sheet metal thicknesses incl. mass optimization
	Validation	Complete
	Number of interfaces to 3rd party	CAD, CAE, Simulation-Database
Spatial Product Analysis (Preparation)	Use case	Automated midsurface generation for simulations
	Validation	Explicit formalization of Modifications and Mappings missing
	Number of interfaces to 3rd party	CAD, CAE, PDM

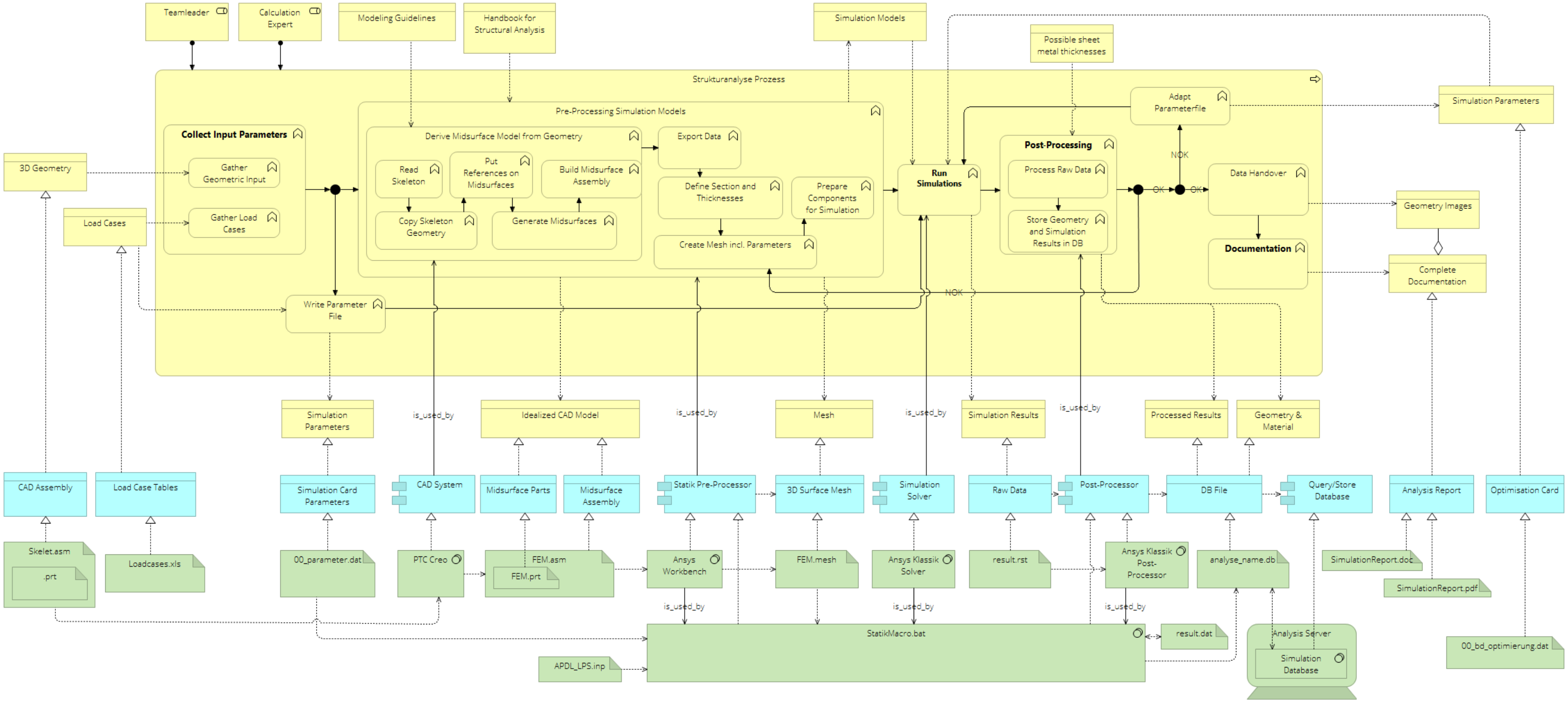


Figure 6-9: Micro-level design process for Case 1 – Structural Analysis.

Table 6-4: Characteristics of identified design automation opportunities for Case 2 – Hydraulic Systems

DA Task Template	Characteristic	Details
Product Architecture Topology Synthesis	Use case	Hydraulic circuit concept generation
	Validation	Simulation models / generic evaluation algorithm missing
	Number of interfaces to 3rd party	ECAD, PDM, CAE
Product Architecture Parameter Synthesis	Use case	Hydraulic component selection for closed circuits
	Validation	Complete
	Number of interfaces to 3rd party	ECAD, PDM, Excel

Table 6-5: Characteristics of identified design automation opportunities for Case 3 – Hydraulic Units

DA Task Template	Characteristic	Details
Spatial Product Architecture Topology Synthesis	Use case	Hydraulic unit piping
	Validation	Fully manual task -> no formalization of knowledge available, but considered possible according to necessary elements. Simulation not necessary
	Number of interfaces to 3rd party	CAD, PDM
Spatial Topology Optimization	Use case	Hydraulic unit block part for open circuits
	Validation	Fully manual task -> no formalization of knowledge available, but considered possible based on necessary elements.
	Number of interfaces to 3rd party	CAD, PDM
Design Configuration	Use case	2D configuration for sales
	Validation	All data available; yet reformatting needed.
	Number of interfaces to 3rd party	PDM, ERP, Excel Model Library, CAD

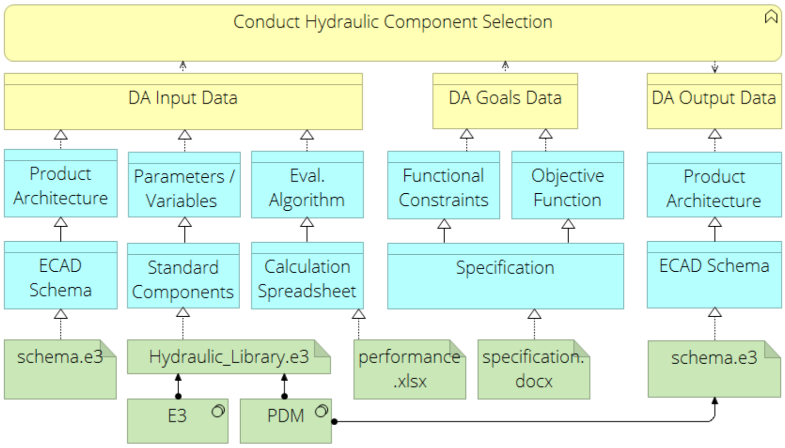


Figure 6-10: Instantiated DA Task Template for automated hydraulic component selection (Case 2)

6.3 Discussion

This section discusses the attained results from the following perspectives: First, the validation of knowledge required for implementation of design automation based on task templates is assessed. In particular, the applicability of the design automation task templates for identification of use cases in industrial environments is critically assessed. Further, the expressiveness and comprehensiveness of the categorization is analyzed. Following this, the applicability of the proposed method for analysis of industrial processes including the supporting tools and technologies is discussed. Especially, the support for identification of redundancies in knowledge formalizations and the potential for design automation integration from a design process as well as tools and technology perspective are investigated. Finally, the implications of the method for identification of design automation use cases for design practice and research are highlighted.

6.3.1 Validation of Design Automation Use Cases

The results depicted above, show that seven use cases for design automation application could be identified based on the design automation task categorization. The discussions among designers when identifying design automation use cases showed that, first, *different levels of granularity* of design automation implementation exist, and, second, the *motivational aspects* of design automation implementation is crucial to determine the right level of detail taken into account by the design automation application. Regarding the *granularity* of design automation tasks, mapping of knowledge to task templates for Case 2 – Hydraulic Systems shows that the identified product architecture topology synthesis task (PATS) for generation of hydraulic circuits can potentially also encompass the product architecture parameter synthesis task (PAPS) for selection of components. Yet, to mitigate economic risk, risk of disappointment due to unfulfilled expectation and to familiarize designers with the concepts of design automation, design automation tasks with a reduced scope should be addressed first and then integrated to more complex design automation applications (Dym and Brown 2012; Willner, Gosling, and Schönsleben 2016). Thus, design automation tasks of reduced scope / complexity can be an entrance point for successive application of design automation in design practice. Therefore, the identification of design automation use cases should consider opportunities on multiple levels of granularity of design automation.

Regarding the *motivational* aspects, the motivation potentially determines the scope and attitude of the design automation application. For example, it can be differentiated whether the intended support aims to generate optimized designs so that designers refine the output based on expert knowledge regarding aspects that are not captured in the support, or the generation of fully validated designs based on comprehensively formalized expert knowledge. Similarly, the design automation goals, e.g. the objective function can only be defined after clarifying the motivational aspects of design automation implementation, e.g. mass vs. cost optimization. Hence, methods are needed to comprehensively estimate the impact of design automation implementation to design practice and enable benchmarking of different scenarios of design automation implementation, cf. Sections 0 and 8.

When considering the *expressiveness* of the design automation task categorization, a comparison of the formalization of input product knowledge for the two potential design automation tasks (PATS & PAPS) yields significant differences, especially regarding knowledge on modifications. For parameter synthesis tasks, modifications refer to the indication of parameters and variables that are subject to change. For topology synthesis tasks, the formalization of modifications accounts for the system boundary. In detail, the interfaces need to be specified including the constraints on potential product architectures. Whereas these details are taken into account in the respective task formalization templates (see Section 9.1), the descriptions of the categories and characteristics do not account for these types of details. Thus, the categorization loses expressiveness at the cost of generality.

6.3.2 Design Automation Integration

The graphical modeling of the design process based on the ArchiMate language enables identification of redundancies readily while modeling the design activities, supporting methods and respective tools and technologies. Generating the process models live on screen during the workshops contributed to the efficiency and effectiveness of the method, since defects in the model can directly be recognized by the participants and resolved so that no more iterations for validation of the generated models are required after completing the first workshop. Even though no detailed information on design activities was prevalent at the beginning of the first round of workshops for each case, the desired design process models are produced in half-day workshops for each case study despite the different design tasks. Regarding the integration of design automation to PLM, the yielded micro-level processes with and without considering design automation show that multiple formalizations need to be taken into account for design automation integration, e.g. see Figure 6-5. Further, the design process models show that the integration of design automation is case specific and needs to be tailored to the design process' supporting methods and technological environment. Hence, the development of a generic design automation toolbox that fits every design process is not realistic. Even though existing toolboxes addressing CAD-based design configuration provide interfaces to multiple CAD systems (Brinkop 2016), the plurality of available information systems supporting design makes the implementation of a generic design automation toolbox that fits every design process not feasible.

Regarding the integration of design automation tasks to workflows, (Ming, Zhenjun et al. 2018) recently proposed *three different scenarios for design automation integration* to design practice that are also observed in this work: original, adaptive and variant implementation of templates. The *first* refers to design automation tasks that require *implementation of a template from the start* each time the design task needs to be automated. For instance, the PATS task as considered for generation of hydraulic concepts requires close integration to the design process. Hence, the structure of the design process alters significantly when comparing the design process including the design automation task to the conventional design process. To enable automation in daily business, the knowledge for automation needs to be formalized in an intuitive manner that reflects current design practice and enables direct computation / automated translation to a computable language. The *second* scenario of design automation task formalization corresponds to tasks that *require adaptation of existing templates* according to changing requirements. However, the majority of formalized knowledge can be reused. For example, the instantiation of a parameter synthesis task needs to be adapted with respect to changes in parameters and variables. Finally, the *third* scenario of design automation task formalization refers to tasks that have a strongly repetitive character and solely the goals are subject to changes in parameters. For example, this is the case for design configuration that uses design parameters as goals for design automation tasks.

Finally, it has to be noted that the validation of the method was conducted for three different case studies within large enterprises. However, in order to generalize the applicability of the method, additional case studies would be needed in small and medium enterprises where the design of similar products potentially takes place on a less regular basis.

6.3.3 Implications for Design Practice

The method proposed in this section contributes to design practice through systematic identification of design automation use cases based on detailed analysis of design processes in industry. The collaborative nature and the descriptive approach enable to take into account different perspectives and increase the designers’ understanding for their design processes. This includes not only the workflows but also the supporting tools and technologies. These finding can be underlined with a qualitative observation for Case 1 when the designers stated that they are surprised by the actual complexity of the design process. This underestimation of the design process is in accordance with the established findings that despite the routine character of design processes often found in industry, they still involve “enough complexity that stakeholders may not fully understand them prior to modeling” (Wynn and Clarkson 2018). Further, understanding the design process is fundamental to improve the design process in design practice (Duffy 2005).

With respect to integration, the templates support specification of interfaces of potential design automation solutions to workflows from both a designer’s and a tools and technology's perspective. In particular, the modeling of possible scenarios of the design process with integrated design automation applications enables communication of the impact of design automation on design practice. Depending on the scenario of workflow integration stated above (original, adaptive and variant), significant effort from the user is required for executing a design automation task. Eventually, design automation rather supports the designer and aims at human-machine collaboration instead of full automation of design (Dym and Brown 2012).

Finally, the analysis of formalized sources of knowledge for implementation of design automation task templates highlights the current level of knowledge formalization in a company. The implementation of the design automation task templates graphically indicates what type of knowledge needs to be additionally formalized to enable design automation. This raises practitioners' awareness with respect to the need for knowledge acquisition and externalization from expert designers.

6.3.4 Implications for Design Research

First, the application of the method contributes to design research by validation of (parts of) the design automation task categorization indicating the appropriateness of the characteristics describing the categorization. Next, the proposed method supports the transition of methods from academia to the industry since raising the designers' awareness of design automation opportunities. Design automation use cases can be readily identified and validated based on the instantiation of task templates.

Even though the method features indication and specification of interfaces to third party software, it lacks details with respect to design automation PLM integration from the perspective of knowledge management. As proposed in Section 5, the formalization of knowledge in a neutral and standardized format (SysML) that is independent on the specific design automation tool potentially enables management of knowledge across different tools and technologies. To avoid duplication of data and integration of formalized knowledge, future efforts should address the perspective of knowledge management in the context of both design automation and PLM.

6.4 Summary

In this section, a five step method for identification of design automation use cases that is based on detailed analysis of design processes as well as implementation and integration of design automation task templates is presented. Application of the method to three industrial case studies shows that the detailed analysis of design processes and supporting tools is fundamental for identification of potential design automation use cases. It not only impacts design practitioners' perception of the design process, but also enables identification of design automation task templates based on design process characteristics. Hence, the answer to *research question two* "*What knowledge is necessary for the identification of design automation use cases in design practice?*" is that detailed knowledge on the design process and the design automation task categorization is necessary to comprehensively identify design automation use cases in practice.

The implementation of the design automation task templates supports the validation of the design automation use case based on assessment of knowledge as well as the identification of redundancies in knowledge formalization. Further, the integration of design automation task templates to design processes enables development of possible scenarios of the design process and specification of interfaces of design automation tools with third party software. Thus, the impact of design automation on design process topologies can be communicated on both a design activity and tools and technology level. However, it has to be noted, that the identification and validation of design automation opportunities is an interactive process that relies on discussions to clarify different viewpoints on the scope of design automation and the availability of knowledge.

Hence, the answer to *research question three* "*How can we best integrate design automation in industrial practice?*" is that design automation needs to be considered case specifically to determine the appropriate level of integration and specify the design automation so that it is comprehensively integrated to the design process. To eventually determine the best opportunity for design automation integration, the motivational aspects of design automation application need to be taken into account which is subject to the following Sections 6 and 7.

These findings of the evaluation of the method can be summarized in a **first contribution:**

A collaborative method for systematic identification and validation of design automation opportunities based on templates that indicate knowledge levels needed for design automation task definition.

Regarding integration of design automation to design practice with respect to design activities and supporting tools and technologies, the **second contribution** of the proposed method can be highlighted:

The development of a method that enables integration of design automation to design practice based on workflow integration and specification of interfaces to third party software.

Building on the design automation task categorization, the application of the method to different industrial use cases implicitly demonstrates the applicability and usability of the design automation task categorization in industrial contexts. Hence, a **third contribution** can be acknowledged as follows:

Industrial validation of the design automation task categorization and the corresponding characteristics that support identification of relevant knowledge for design automation implementation.

The following sections address the aforementioned need for systematic estimation of design automation application in industrial practice. First, the identification of appropriate metrics to communicate the value of design automation application is identified in Section 7. Following this, the method for estimation of the design automation potential in design practice is introduced in Section 8.

7 Method for Derivation of Metrics for Impact Estimation of Design Automation

After successful identification of design automation use cases in design practice, the second step of the methodology refers to the derivation of metrics for estimation of the impact of design automation on design practice. Despite the history of research and application of design automation for more than four decades (J. Panchal et al. 2015), only few works have focused on the assessment of design automation potential for design performance improvement prior to its implementation or comprehensive a-posteriori validation of success of an implementation. Hence, there is a substantial lack of methods for estimating the potential impact of design automation on design practice (Verhagen et al. 2015), but also the lack of metrics that enable quantification of the related design automation potential when used in industry. Industrially validated metrics reflecting practitioners' needs are also necessary to guide, benchmark and assess the development of new design automation methods. To enable comprehensive assessment of the design automation potential, metrics need to account for the design process including its technological environments.

To address the lack of metrics, Section 7.1 introduces a method for derivation of metrics. The method builds on the method for identification of design automation use cases as well as potential failure modes that can possibly be avoided by application of design automation. In Section 7.2, the results of application of the method to two industrial cases are presented indicating the yielded metrics that enable estimation of the impact of design automation on design practice. Then, Section 7.3 critically assesses the results yielded from industrial evaluation and finally, Section 7.4 presents a summary of the section and points out the contributions of the method addressing research question 4 as well as some concluding remarks.

7.1 Method

In response to the need for metrics, a top-down method for the derivation of metrics for estimation of the impact of design automation on design practice is proposed. To enable a top-down strategy (Koziolek 2008) for derivation of metrics, the method builds on the analysis of design processes and identification of design automation goals that are to be measured. For identification of design automation goals, the proposed method builds on failure modes analysis in the design process. This is based on an assumption that if design automation is used within a design process some of the failure mode causes are removed and the effects are mitigated. Thereby the method contextualizes design automation and design practice based on a failure modes analysis reflecting the designers' needs. This enables to include the GQM paradigm in the top-down approach proposed in this work.

The main steps that coin the method for metrics derivation for potential estimation of design automation are depicted in Figure 7-1. The yellow layer depicts the main activities, exchange of information as well as involved stakeholders. Whereas the blue layer indicates the applications of design process support methods, the green layer denotes the tools that were actually used in this work. It can be seen that the identification of design automation use cases as presented in Section 6 constitutes a major element of the method for derivation metrics for impact estimation of design automation (Steps 1,2 and 5). Hence, the method that is presented in the following will systematically reuse elements that have already been introduced in Section 6.

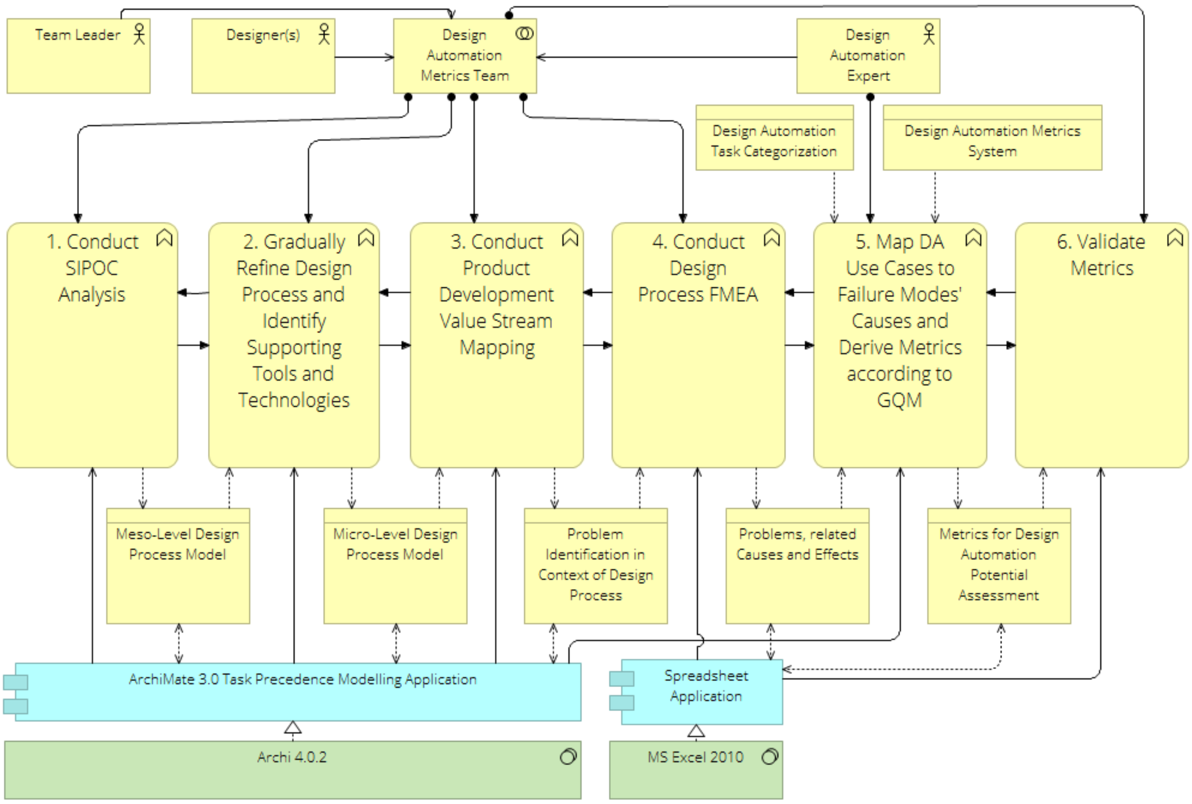


Figure 7-1: Workflow of the proposed method: Main steps (central layer, yellow), team composition and external information (top layer, yellow) as well as supporting tools and technologies (blue & green).

The following points present the different steps of the method for derivation of metrics in more detail:

1. In analogy to step one described in Section 6.1, a SIPOC analysis is conducted to capture the design process on the meso-level.
2. In analogy to step two described in Section 6.1, the information of the SIPOC analysis is gradually refined and enriched to capture the design process on the micro-level to map design activities and supporting tools and technologies.
3. Product development value stream mapping (McManus 2005) is applied for identification of possible failure modes in the micro-level ArchiMate model of the

design process. Possible failure modes are mapped with respect to the activities and related information flows as conducted in McManus (2005), but also associated to tools and technologies.

Regarding the value stream mapping, the following types of information wastes are considered as proposed in McManus (2005):

a. **Waiting**: Late or early delivery causing waiting times,
b. **Inventory**: Issues with data, e.g. complicated retrieval, outdated/redundant information etc.,
c. **Over-Processing**: Tedious formatting and post-processing of data,
d. **Over-Production**: Too much information is produced and distributed,
e. **Transportation**: Communication issues, data incompatibility,
f. **Unnecessary motion**: Physically or between tools to process data,
g. **Defective products**: Errors in data, information, reports.

To additionally support failure mode identification, questions are asked for each activity related to the categories of failure modes as defined by Chao and Ishii (2007):

a. The availability and quality of **knowledge** to conduct an activity,
b. The quality of **communication** of information for each activity,
c. The requirements and information needed for accurate **analysis** of information processed in each activity,
d. The quality of **execution** of an activity,
e. The rate of **change** of information as well as the degrees of freedom for each activity,
f. The maturity of **organization** of a design activity.

Finally, participants are also asked to brainstorm for possible failure modes that can occur in the design process. Once a possible failure mode is identified, it is integrated to the design process model according to the syntax of the meta-model depicted in Figure 4-2. Thus, identified failure modes can be associated directly to the related design activity, tool or technology. Figure 7-2 shows the micro-level process model of the example introduced in Section 6, Figure 6-3 including three identified failure modes highlighted in purple and directly associated to the respective artefact in the model.

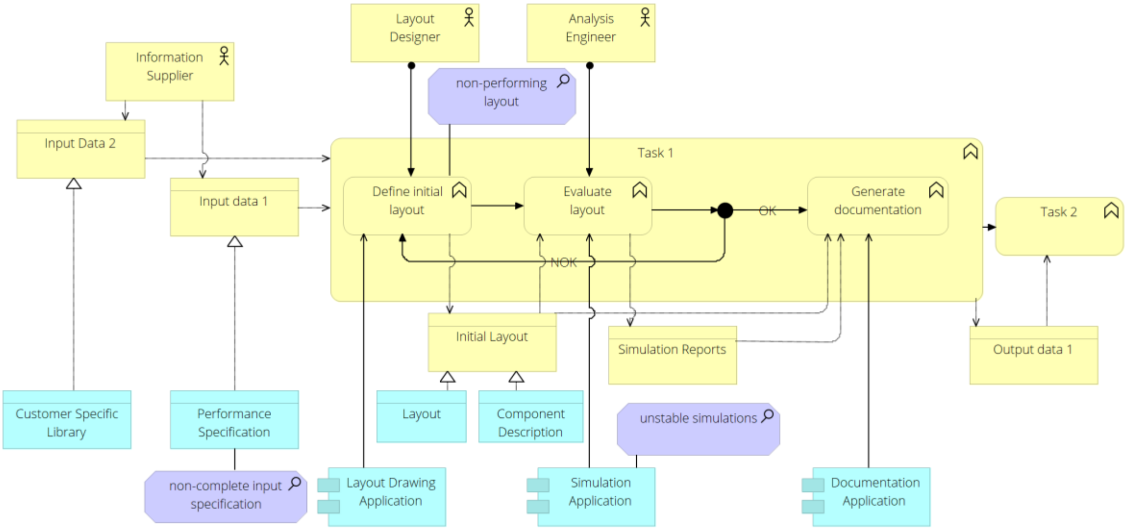


Figure 7-2: Results of design process value stream mapping integrated to the micro-level process model for the example introduced in Figure 6-3.

4. A Design Process Failure Mode Effects Analysis (dpFMEA) (Chao and Ishii 2007) is performed for structured identification of the effects and causes of potential failure modes that can be used for determination of metrics according to GQM. Possible failure modes identified in Step 3 are listed within a table in the sequence as occurring in the process. For each failure mode, first, the corresponding effect is listed. Next, one or more causes of the potential failure modes are identified. Despite being useful for prioritizing actions to be taken for adaptation of the design process, the remaining steps of dpFMEA such as qualitative rating of importance, occurrence and detection are considered out of scope for this study since focus is put on metrics derivation. Table 7-1 details the cause effect analysis for the failure mode “layout does not pass evaluation” and the activity “define initial layout” from the example shown in Figure 7-2.

Table 7-1: Excerpt of FMEA table for the example introduced in Figure 6-3.

Activity	Failure Mode	Effects	Causes
Define initial layout	Layout does not pass evaluation	Additional iterations, sub-optimal layouts, lack of design space exploration	Lack of time for investigation of alternatives
			Faulty specification

5. The structured listing of potential failure modes, effects and causes as obtained from the dpFMEA is used for derivation of metrics for assessment of the failure modes’ causes that are potentially resolved or mitigated through implementation of design automation, cf. design automation goals in Figure 4-2. Implicitly, this step applies the steps 3 – 5 from the method for identification and integration of design automation use cases (Section 6.1) so to identify which failure modes’

causes are potentially affected by the implementation of design automation. Thereby, the mitigation of causes refers to the design automation goals and information on the effects can be used as questions to guide metrics definition related to quantification of the impact of design automation on design practice. The design automation goals are also mapped to the list of design automation drivers and software quality categories in Table 2-1 to Table 2-5. This enables identification of the related metrics. Since substantial knowledge on design automation is required for the identification of the use cases, this step is to be conducted by the design automation expert who is also part of the workshop team. Table 7-2 illustrates the identification of metrics based on failure mode causes, effects and related design automation drivers for the case that product architecture topology synthesis is implemented to support layout generation for the example shown in Figure 7-2.

Table 7-2: Example of identification of metrics to mitigate a specific failure mode cause.

Activity	Failure Mode	Effects	Causes	DA Driver	Metrics
Define initial layout	Layout does not pass evaluation	Additional iterations, sub-optimal layouts, lack of design space exploration	Lack of time for investigation of alternatives	Generation of alternatives	# of evaluated alternative designs per design activity / average # of solution alternatives investigated per activity
				Solving complex design tasks	# of parts with errors / total # of lists of parts produced
				Enable the development of customer specific solutions	# of product failure modes of products delivered within year XY / # of delivered products within XY
				Engineering First Pass Yield (FPY)	# of FCs designs submitted w/o rejection / Total # of FCs
				Cost reduction	Number of iterations in process / average number of iterations
				Cost reduction	Total time for conducting parameter synthesis activity with / w/o DA

6. The industrial validation of case study specific metrics is performed based on a tabular listing of identified metrics that are yielded in Step 5. For validation, the metrics are organized in a table format as outlined in Table 7-3, which contains information with respect to following aspects:
    - ◊ The addressed design automation task in relation to the identified failure modes.
    - ◊ The relevant design automation driver and software quality dimensions according to Table 2-1 - Table 2-5 to indicate the motivational aspect of design automation implementation.

◊ A metric description indicating what is measured.

Based on this information, the workshop participants are asked to rate each metrics with respect to:

◊ Metric measurability for assessment of whether the relevant data can be collected in the given context (Yes/ No)
◊ Data availability as an indication if data sets already exist for the defined metric (Yes/ No)
◊ Metric relevance as an indication of whether the metric is valid and explicit in the context used (Yes/ No)

In the workshop, the yielded table is gradually discussed until the participants agree on the assessed criteria of each metric. For the case that a metric is considered measurable, the available data collection method as well as the implementation to actually evaluate the metric based on the data is indicated to provide the information required for implementation. Regarding data collection, focus is put on data collection based on corporate IT systems, e.g. ERP, PDM. If IT systems do not offer the required level of detail of information, e.g. for time management or intangible metrics such as motivation, work sampling (Robinson 2009) is proposed as a means to increase the resolution of data collection based on individual assessment of engineers.

For the case that metrics are missing, another iteration of Step 5 is conducted to refine the list of metrics provided.

Table 7-3: Table for conducting the validation Step 6. It summarizes metrics including the quantitative descriptions yielded in Step 5 and enables practitioners to rate each metric with respect to 'measurability', 'data availability' 'relevance' and its 'origin'. The very last column is used to indicate whether the metric is newly defined or taken from Table 2-1 - Table 2-5.

Design Automation Task	Metric Category	Metric Description and Implementation	Measurability	Data Availability	Relevance	Origin

Workshop Organization

For conducting the method in industrial practice, in total three collaborative workshops are needed. The first workshop aims to introduce the method and conducting the method's Steps 1 to 3. Steps 1 and 2 are conducted in analogy to the workshop for the method for identification of design automation opportunities requiring the same skills of participants (cf. Section 6.1). Whereas the second workshop pursues Step 4 to analyze failure modes, their causes and effects, the third workshops conducts Step 6 to industrially validate the metrics yielded in Step 5. Step 5 for identification of design automation use cases and metrics derivation should be conducted by a design automation expert knowledgeable about the design automation opportunities as indicated in the design automation task categorization, cf. Section 5. In order to guide workshops and introduce necessary parts of applied methods, one participant needs to be familiar with the concepts of design process value stream mapping (Workshop 1, Step 3) and design process FMEA (Workshop 2, Step 4). Similarly to the workshop setup for the method for identification of opportunities, a mixed team consisting of at least two designers and the team leader should be selected to account for potential failure modes on multiple levels of abstraction of the design process and rate metrics from different viewpoints of practitioners.

7.2 Results

Application of the method for derivation of metrics for design automation assessment was conducted for the industrial test cases 1 and 2 (see Section 4.3). In the following, the details of the workshop setup are introduced and the results yielded in the workshops are presented.

7.2.1 Workshop Setup

Three collaborative workshops were conducted for each case. It has to be noted, that parts of the first workshop (Steps 1 and 2) overlapped with application of the method for identification of design automation use cases, see Section 6.2. For each workshop, the thesis author first presented and introduced the respective methods as well as the supporting concepts relevant for the workshops. Also, the thesis author was in charge of the role of a design automation expert to conduct Step 5. For all workshops, a working colleague of the thesis author supported by documenting captured information in ArchiMate, MS Excel, respectively. Table 7-4 provides the overview of workshop participants and also indicates which participant contributed to which workshop.

Table 7-4: Workshop participants

Participant	Position	Work experience (years)	Workshop attendance
CASE 1 – Structural Analysis			
Designer 1-A	Designer, Team leader	4	1,2,3
Designer 1-B	Design Engineer	10	1,2,3
Designer 1-C	Department Manager	12	1,3

Participant	Position	Work experience (years)	Workshop attendance
Designer 1-D	Design Engineer	8	2
Designer 1-E	Design Engineer	6	2
CASE 2 – Hydraulic Systems			
Designer 2-A	Designer, Team leader	11	1,2,3
Designer 2-B	Designer, Team leader	11	1,2,3
Designer 2-C	Department head	35	1,2,3
Designer 2-D	Designer	4	2,3

7.2.2 Workshop Results

In the following, the corresponding results of the application of the method for metrics derivation for potential estimation of design automation (as given by Figure 7-1) are successively introduced. The results of the meso-level analysis as conducted in Step 1 are already presented in Section 6, Figure 6-6 and Figure 6-7. Similarly, results of the micro-level analysis of design processes for Step 2 are indicated in Table 6-3 and Table 6-4. With respect to the identification of possible failure modes, the value stream mapping was applied and possible failure modes are mapped directly to the design process model. To indicate the results, the number of possible failure modes is listed in Table 7-5. Within Step 4, for each possible failure mode, one or more causes are listed in a dpFMEA table as well as the possible effects. The numbers of related causes are depicted in Table 7-5. Further details on Step 3 and 4 can be found in the appendix. Regarding the design automation use cases as identified within Step 5, the design automation tasks according to the design automation task categorization as well as the corresponding use cases are listed in Section 6 within Table 6-3 and Table 6-4. To provide a measure for efficiency of the method, Table 7-6 indicates the required efforts based on the time spent for each workshop. Finally, the metrics that were derived in Step 5 and validated in Step 6 are organized according to the structure depicted in Table 7-3. The resulting metrics descriptions for Cases 1 and 2 are listed for the addressed design automation tasks in Table 7-7 and Table 7-8, respectively. The metrics are organized according to the associated drivers. For the case that a metric is relevant for more than one design automation task, the metric is listed one time only with all relevant design automation tasks listed in the column indicating the design automation task.

Table 7-5: Results of application of the method for derivation of metrics for design automation assessment

Key Characteristic	Case 1	Case 2
Number of additional activities identified in Step 2	17	25
Absolute time for workshop 1	4 h	3 h
Number of identified possible failure modes in the design	16	9

Key Characteristic	Case 1	Case 2
process		
Number of identified causes related to possible failure modes	79	28
Total time for workshop 2	3 h	1.5 h
Considered design automation use cases for metrics derivation	Spatial Product Architecture Parameter Synthesis for automated sizing of sheet metal thicknesses incl. mass optimization	Product Architecture Topology Synthesis for hydraulic circuit concept generation, Product Architecture Parameter Synthesis for hydraulic component selection
Addressed failure modes / causes by design automation implementation	4 / 13	4 / 8
Total time for workshop 3	1 h	1 h

Table 7-6: Absolute time required for the workshops for Case 1 and Case 2, respectively

Workshop	Absolute Time Case 1 [h]	Absolute Time Case 2 [h]
1	4	3
2	3	1.5
3	1	1

Table 7-7: List of metrics including metrics description and information required for quantitative evaluation in practice for Case 1 – Structural Analysis. Further, the table indicates measurability (meas.), data availability (dat.av.), relevance (rel.) per metric. The very right column ("Origin") indicates whether a metric is newly derived ("N"), adapted ("A") or reused ("R") based on Table 2-1 to Table 2-5. Metrics not considered relevant by practitioners are written in grey font. Metrics considered relevant, and metrics which adapt or extend the initial metrics system are highlighted in bold letters.

DA Task	DA Driver	Metric Description and Implementation	meas.	dat.av.	rel.	Origin
Automated sizing of sheet metal thicknesses	**Cost reduction**	**Total time for conducting parameter synthesis activity with / w/o DA** **Data Measurement: IT System; Implementation: Check-in/ Check-out time of models**	**x**	**x**	**x**	**A**
		Number of iterations in process / average number of iterations **Data Measurement: IT System; Implementation: Number of revised models per project**	**x**		**x**	**N**
		Cycle time: Lead time per iteration of parameter synthesis Data Measurement: IT System; Implementation: Checkin/-out per revision	x		x	R
		Lead time for sub-activity (simulation) for parameter synthesis Data Measurement: IT System; Implementation: Log data of simulation application	x		x	R

DA Task	DA Driver	Metric Description and Implementation	meas.	dat.av.	rel.	Origin
Automated sizing of sheet metal thicknesses		On-time delivery: Number of Functional components released on time / Total number of Functional Components Data Measurement: IT System: Implementation: Plan vs. release	x		x	R
		% of time spent for R&D activities / project				R
	Error reduction in design	# of missing design guidelines				N
		# of lacking check procedures				N
		# of Engineering changes for design using DA / w/o DA in average / project	x	x		A
		# of rework rate due to defects in input			**x**	**N**
	Generation of optimized design	**performance manual design vs. DA design** **Data Measurement: IT System; Implementation: model comparison**	**x**		**x**	**N**
	Establishment of a knowledge base / Capitalization of knowledge	Time for teaching new employees on the job with / w/o DA	x			N
	Enhanced manufacturability	**Cost of manufacturing for design with / w/o DA** **Data Measurement: IT System; Implementation: model comparison**	**x**	**x**	**x**	**N**
	Enable the development of customer specific solutions	**# of validated use cases / total # of use cases** **Data Measurement: IT System; Implementation: model documentation vs. specification**	**x**	**x**	**x**	**N**
	Generation of alternatives	**# of evaluated alternative designs per design activity / average # of solution alternatives investigated per activity** **Data Measurement: IT System; Implementation: number of stored variants**	**x**	**x**		**A**
	Quality in use: Effectiveness	**Number of designs generated using DA / # of generated designs** **Data Measurement: IT System; Implementation: flag in designs generated by DA**	**x**		**x**	**N**
	Quality in use: Efficiency / Cost-effectiveness	**Computational Expenditures per project / average personal cost per project** **Data Measurement: IT System; Implementation: log-data**	**x**	**x**	**x**	**N**
		Personal Expenditures per project / average personal cost per project **Data Measurement: IT System; Implementation: time management**	**x**	**x**	**x**	**N**

Table 7-8: List of metrics including metrics description and information required for quantitative evaluation in practice for Case 2. Further, the table indicates measurability (meas.), data availability (dat.av.), relevance (rel.) per metric. The very right column ("Origin") indicates whether a metric is newly derived ("N"), adapted ("A") or reused ("R") based on Table 2-1 - Table 2-5. Metrics not considered relevant by practitioners are written in grey font. Metrics considered relevant, and metrics which adapt or extend the initial metrics system are highlighted in bold letters.

DA Task	DA Driver	Metric description and implementation	Meas.	Dat.av	Rel.	Origin
Hydraulic circuit concept generation	Reuse of knowledge / Reuse of existing solutions & designs	**# of designs validated by integration of field data / # of designs generated** **Data Measurement: IT System; Implementation: logon history**	**x**	**x**	**x**	**N**
	Establishment of a knowledge base / Capitalization of knowledge	# of concepts discussed in team / # of concepts discussed on average			x	R
		# of stored designs / # of designs created Data Measurement: IT System; Implementation: CAD log	x	x	x	R
	Enable the development of customer specific solutions	# of product failure modes of products delivered within specific duration / # of delivered products in specific duration Data Measurement: IT System; Implementation: engineering change log vs. release data	x	x	x	R
	Solving complex design tasks	Ideation Quantity: Total number of generated ideas				R
		Ideation Variety: Total Number of unique ideas				R
		Ideation Quality: Measures the feasibility of an idea and whether it meets the design requirements			x	R
		Is creativity in design improved?				N
	Generation of alternatives	**# of evaluated alternative designs per design activity / average # of solution alternatives investigated per activity**			**x**	**A**
	Quality in use: freedom from risk	**Is the safety in the design process improved**			**x**	**N**
	Increase Engineering Design Data Quality / Data Accessibility	Time spent for scanning database for right item / total activity time Data Measurement: IT System; Implementation: log data vs. time management	x	x	x	R
		number of data items relevant to the user's task and accessible / total number of data items relevant to the user's task Data Measurement: IT System; Implementation: database analysis	x	x	x	R
	Cost reduction	total expenditures per project / total average cost per project Data Measurement: IT System; Implementation: cost evaluations based on ERP and PDM data	x	x	x	R
		Schedule Performance Indicator : Budgeted cost of work performed [€] / Budgeted cost of work scheduled [€] Data Measurement: IT System; Implementation: comparison of actual labour cost to planned cost	x			R
		Total time for conducting design task Data Measurement: IT System; Implementation: time	x		x	R

DA Task	DA Driver	Metric description and implementation	Meas.	Dat.av	Rel.	Origin
Hydraulic circuit concept generation / Hydraulic component selection for closed circuits		management evaluation				
		Number of iterations in process / average number of iterations Data Measurement: IT System; Implementation: number of revised models per project	x	x		N
		On-Time Delivery: Number of FCs released on time / Total number of FCs Data Measurement: IT System: Implementation: plan vs. release	x	x	x	R
		Cycle time: Lead time per iteration of parameter synthesis Data Measurement: IT System; Implementation: Checkin/-out per revision	x			R
		% of time spent for R&D activities / project **Data Measurement: IT System; Implementation: time management evaluation**	**x**		**x**	N
		% of relevant ideation activities regarding product, process and other domains by any individual			x	R
	Increase of productivity	Employee satisfaction level Data Measurement: Work sampling; Implementation: Likert scale rating			x	R
		% of activities done with lower motivation and irrelevant activities Data Measurement: Work sampling; Implementation: based on information for each activity				R
		Delay times: Mean and deviation, or distribution, of wait times (best) Data Measurement: IT System; Implementation: time management evaluation	x		x	R
	Error reduction in design	**# of engineering changes for specific design / engineering changes for designs on average** **Data Measurement: IT System; Implementation: engineering change log**	**x**	**x**	**x**	**A**
		# of rework rate due to defects in input **Data Measurement: IT System; Implementation: time management evaluation**	**x**		**x**	**N**
	Generation of optimized design	Performance of design using DA / w/o DA Data Measurement: IT System; Implementation: benchmark cases	x	x		N
		Key performance indicator of product / performance according to specification **Data Measurement: IT System; Implementation: model analysis vs. specification**	**x**		**x**	**N**
	Establishment of a knowledge base / Capitalization of knowledge	Time for teaching new employees on the job with / w/o DA				N
	Enable the development of customer specific solutions	# of validated use cases / total # of use cases Data Measurement: IT System; Implementation: simulation log files	x	x		N
	Quality in use: Freedom from risk / Economic risk mitigation	**ROI: yield of time savings / investment cost for implementing DA**			**x**	**A**

DA Task	DA Driver	Metric description and implementation	Meas.	Dat.av	Rel.	Origin
	Quality in use: Effectiveness	Number of designs generated using DA / # of generated designs				A
	Quality in use: Efficiency / Cost-effectiveness	Computational Expenditures per project / average personal cost per project				A
		Personal Expenditures per project / average personal cost per project			**x**	**A**
Hydraulic component selection for closed circuits	Generation of optimized design	**Cost for design using DA / w/o DA** **Data Measurement: IT System; Implementation: benchmark cases**	**x**		**x**	**N**
	Error reduction in design	Engineering first pass yield: # of FCs designs submitted w/o rejection / Total # of FCs			x	R
	Reuse of knowledge / Reuse of existing solutions & designs	Ratio number of standard parts in product design / total number of parts in product design Data Measurement: IT System; Implementation: model comparison in relation to product platform	x	x	x	R
		The percentage of engineered parts / components within a finished product used in at least one other previously finished product Data Measurement: IT System; Implementation: model comparison in relation to model database	x	x	x	R

7.3 Discussion

This section first discusses the results obtained from application of the method for metrics derivation. In particular, qualitative observations made during the workshops are used to assess the method's usability and applicability in industrial contexts. Second, an evaluation is conducted to assess the usefulness of the proposed method by analyzing the metrics that are attained within the workshops as well as the resulting design automation metrics system in general. Finally, the implications of this research for both design automation practice and research are highlighted.

7.3.1 Application Evaluation

The evaluation of the method was conducted for two different types of design tasks that are part of large-scale designs, first for cranes, and second, for heavy construction machinery. Whereas the first accounts for a design task with strictly bound design space by standards and regulations, the second features more design freedom. Thus, there is a difference in identified use cases for design automation application (cf. Section 6).

Within Table 7-5, it can be seen that the number of identified causes for Case 1 exceeds almost three times the number of causes for Case 2. The explanation behind this is partly due to the substantially less time that was invested for failure modes cause identification for Case 2 workshop (as indicated in Table 7-6) as well as the different characteristics of the design tasks and workshop participants' profiles and backgrounds. Further, it was observed qualitatively, that the identification of failure modes was biased by the presence of department managers. In contrast to Case 1 workshops, this influenced the designers starting to discuss possible failure modes in detail rather than continuing brainstorming for potential failure mode that are relevant for their work. On the other hand, the application of structured questions with respect to potential failure modes as indicated in Section 7.1 showed to encourage participants in identification of additional potential failure modes.

Regarding the efficiency of the method, it can be seen in Table 7-6 that less than 8 hours were needed for analysis of a design process. Even though direct comparison of results is impossible, this is substantially less than other reported efforts for design process analysis, e.g. (Schut, Kosman, and Curran 2013) who report “three weeks to capture and analyze the problems and process steps”. Potentially, the interactive approach of live modeling of design process and mapping of problems bears some aspects of game playing which contribute to enhanced efficiency. In fact, it proved useful to ask practitioners to alternate in reporting potential failure modes so that all participants are involved and some dynamics is created. This observation is well aligned with findings in literature that games support communication between stakeholders and mitigate hierarchical relations (Brandt and Messeter 2004).

The validation of the yielded metrics was conducted with respect to team workshops until the workshop participants coming from the industry agreed on a shared opinion. However, this approach potentially biases individual perspectives of relevance and measurability, in particular, when considering the influence of senior designers’ opinion on less experienced team members. This could be resolved by means of individual rating of metrics and subsequent team discussion to enable systematic consideration of various viewpoints of the team members.

7.3.2 Success Evaluation

Table 7-7 and Table 7-8 show that different sets of metrics were attained for the two cases. This is due to specific failure modes and causes for each of the Cases. For Case 2, substantially different metrics are of interest for product architecture topology synthesis than for the parameter synthesis tasks. In particular, metrics related to the generation of alternative designs or safety in design is considered relevant for the synthesis of topologies. On the other hand, a similar set of metrics can be identified for Cases 1 and 2 with respect to the parameter synthesis tasks that potentially can be implemented for both cases. The task that focuses on determination of structural parameters and hydraulic component selection both consider metrics related to design process efficiency and effectiveness important, in particular related to generation of optimized designs. On the other hand, Case 2 focuses on aspects related to reuse of knowledge to support standardization whereas Case 1 aims at further reducing costs through taking into account manufacturing constraints. Thus, it can be argued that for each type of design automation task category, a different set of metrics can be identified to serve as a starting point for metrics derivation. However, additional metrics need to be derived tailored to the particular use cases and interests of practitioners.

Further, the selection of relevant metrics strongly depends on the experience designers have with computational methods and design automation. The hydraulics department (Case 2) did not take into account the computational costs that are likely to increase substantially through implementation of design automation. On the other hand, the structural analysis department that relies on computationally expensive finite element simulations considers computational expenses relevant. Thus, the application of a standardized set of metrics can also be used to avoid missing metrics that are crucial for comprehensive assessment of design automation in practice.

Considering the metrics in more detail, for all types of design automation use cases, metrics related to cost and error reduction are considered of relevance from the practitioners' point of view. This is well aligned with industrial surveys aimed at ranking drivers for design automation implementation (Amen, Rask, and Sunnersjö 1999; Cederfeldt, Elgh, and others 2005; Rigger and Vosgien 2018). On the other hand, the online survey conducted by Rigger and Vosgien (2018) that contained a qualitative rating of relevance of design automation drivers, suggested that generation of design alternatives is considered the least important driver for design automation implementation from the practitioners' perspective. This is contradictory to the results of this work, where the number of investigated design alternatives and ideation quality are considered important metrics for design automation in the early phases of design. This underlines the findings already reported in literature that design practitioners lack the full perspective of design automation opportunities (Rigger and Vosgien 2018; Chakrabarti et al. 2011).

From the metrics derived in this work as well as the ones collected through the comprehensive literature review in Section 2, it can be seen that the engineering design community lacks standards for design performance assessment as opposed to the software quality assessment that started developing standards already in the 1990s (ISO/IEC 2011). Thus, multiple definitions of similar metrics for design performance assessment can be found in literature. Still, metrics are missing with respect to some design automation drivers, e.g. "solving complex design tasks". The application of the proposed methodology is shown to enable derivation of metrics specific to design automation and is extending the set of generic design KPIs available in literature. The derivation of metrics showed that also metrics that violate the NEAT guidelines for metric definition (Duffy 2005) are potentially of practical relevance. In particular, when it comes to assessment of intangible aspects (Škec, Cash, and Štorga 2017) that involve subjective rating, e.g. of "safety", or "satisfaction", which are considered relevant for Case 2. Hence, subjective metrics also need to be taken into account for comprehensive assessment of design automation in practice.

With respect to the rating of metrics, it can be observed that subjective metrics and metrics related to investigation of alternatives designs are considered to be not measurable. Hence, all metrics considered measurable are based on data collection from IT systems. Potentially, this is due to a lack of awareness of work sampling measurement methods as introduced in (Škec, Cash, and Štorga 2017) as well as lack of systematics in data management. The latter causes that the design history is not documented comprehensively with respect to investigated design alternatives. This is also confirmed from the point of view of “availability of data”: the results show that data is available related to project information from an ERP or data that can be directly measured from product characteristics such as its performance or usage of standard parts. Regarding the relevance of the proposed metrics, the rating strongly depends on the team characteristics and also whether a metric is valid or true in a specific context. However, it has to be noted that metrics are validated on a conceptual level, only. More detailed empirical validation of the proposed metrics needs to be conducted with respect to actual measurements in design practice.

Only a few metrics from the standards for software quality assessment could be reused since they are specific to the assessment of implemented software. Still, these metrics are needed for integrated assessment of impacts of design automation implementation, e.g. the impact of design automation implementation on data accessibility as considered relevant for Case 2. Further, the presented metrics system can also be reused for design automation success evaluation once a tool is implemented.

7.3.3 Implications for Design Practice

The derivation of metrics for design automation as presented in this work contributes to design practice through building the foundation for systematic and quantitative assessment of the impact of design automation implementation on design practice for specific industrial cases. The focus on involving the practitioners enables raising the awareness of design automation opportunities in practice and also fosters the communication of design automation opportunities to support design practice. The derivation of metrics based on failure modes as well as the corresponding causes that are potentially mitigated by design automation further supports convincing practitioners of the benefits design automation implementation potential offers. Additionally, the comprehensive assessment of design automation based on derived metrics goes beyond current practices of assessing time savings achieved by automation. Thus, not only decision makers in industry but also the practitioners can be convinced of the potential benefits. Successive application of the method can be used to gradually enrich the general library of metrics in order to yield a standardized set of metrics. This metric library is applicable to potential estimation of design automation but is also for design performance assessment in general.

7.3.4 Implications for Design Research

Since specific methods can be derived for each design automation category, this enhances research on generic design automation methods to consider practitioners' needs independent from specific use cases. Therefore, the establishment of the design automation metrics library that is yielded through successive application of the method supports communication of industrial needs to academia. On the other hand, as indicated above, the method supports raising practitioners' awareness of the design process they are involved in and the corresponding design automation use cases in relation to the identified failure modes. Consequently, this method also supports the transition of methods from academia to the industry.

7.4 Summary

In this section a 6 step method for top-down derivation of metrics for impact estimation of design automation is presented. It builds on the method for identification of design automation use cases and detailed analysis of design processes. Thereby, the method relies on established methods for process analysis and successfully links them to the context of design automation application. In particular, the analysis of failure modes as well as the corresponding causes enables to contextualize design automation and design practice based on design automation goals that mitigate failure modes causes. The two industrial cases demonstrate that the proposed method enables the derivation of a set of metrics for estimating the potential of design automation.

The collaborative character of the workshop organization facilitates the derivation of metrics. In particular, the application of a standardized graphical modeling language and the live modeling in workshops has contributed to efficient and effective documentation and analysis of design processes. The involvement of practitioners supports raising the awareness for design automation opportunities and potentials for improvement. For the two industrial cases, a similar set of metrics is yielded for design automation tasks belonging to the same type of design automation task category. Hence, research question four: "*What are appropriate metrics to assess the impact of design automation on design performance*" can be answered as follows:

The impact of design automation on design performance needs to be specifically addressed for each design process and design automation task. However, for similar design automation tasks, potentially similar metrics can be used for comprehensive assessment of the impact of design automation implementation. The application shows that metrics along the dimensions of efficiency, effectiveness, knowledge and quality in use are of relevance for practitioners.

Thus, the **first contribution** can be highlighted as follows:

A collaborative method that contextualizes design automation and design practice based on failure modes and enables derivation of metrics tailored to the design process and design automation task.

In the long term, successive application of the method to different industrial case studies enables derivation of a standardized set of metrics for each design automation task category. This enables research on design automation methods to put the focus directly on the needs of practitioners, which are reflected by the metrics. Further, benchmarks for design automation methods can be developed for each type of design automation task category based on practitioners' needs.

With this respect, two additional contributions can be acknowledged:

Second contribution: *The establishment of a design automation metrics system that features reuse.*

Third contribution: *The link of metrics and design automation task categorization to enable systematic reuse of metrics for potential estimation of design automation.*

However, regarding the organization of the yielded metric system, in analogy to the software quality standards, a rigorous categorization of metrics dimensions is needed for systematic organization of metrics. Further, future research should also account for the specific benefits the possible design automation methods offer. Hence, informed decisions with respect to design automation implementation and selection of a design automation method are enabled.

8 Method for Estimation of the Impact of Design Automation on Design Practice

After the identification of design automation use cases and derivation of metrics, the third step of the methodology for design automation task definition accounts for the quantification of design automation potential based on the rating of a selected set of metrics. However, only few methods exist addressing the systematic assessment of potential impacts of design automation on design practice. What is more, these methods do not consider both the design automation task and the design processes for potential estimation of design automation. Focus is often put on the type or characteristic of a design automation task (Emberey et al. 2007; Mulder et al. 2015; van der Velden, Bil, and Xu 2012; Pal and Ghosh 2017), or the assessment of design processes in general (Schut, Kosman, and Curran 2013; Verhagen et al. 2015). Hence, a combined approach that considers the perspectives of design processes as well as design automation tasks is needed. Particular focus needs to be put on the actual quantification / estimation of metrics to assess the impact of design automation implementation.

In response to this need Section 8.1 introduces a method addressing the need for systematic estimation of the potential impact of design automation on design practice. This is based on parts of the methods for identification of use cases and derivation of metrics as well as design automation task templates. With respect to the latter, the method aims at systematic reuse and integration of metrics. Reuse is based on design automation task templates that account for the motivational aspects of design automation implementation, cf. Section 5.3.2. Section 8.2 presents the results yielded for qualitative potential estimation for two industrial cases, in particular Case 1 – Structural Analysis and Case 2 – Hydraulic Systems, see Table 4-1. Thereby, results described in Sections 6.2.2 and 7.2.2 are systematically reused and further elaborated on to systematically assess potential estimation of design automation in industrial contexts. Section 8.3 discusses the attained results from the perspective of both research and industry. Finally, Section 8.4 provides a summary indicating the contributions, in particular, with respect to research question 5.

8.1 Method

The method presented in this section is based on the findings of Section 7. Similar metrics are relevant for the same type of design automation tasks independent on the application domain. Hence, the design automation task templates that account for the motivational perspective of design automation implementation (see Section 5.3.2) can be enriched with metrics yielded in industrial case studies. Thus, the templates enable reuse of metrics by readily providing a list of metrics when integrating a design automation task template to the design process model. The list of metrics can then be refined and filtered with respect to the investigated design process, design domain as well as corporate standards. Collaborative workshops are then used to account for different viewpoints for the selection and estimation of metrics' values. The four main steps of the method are depicted in Figure 8-1 and are characterized as follows:

1. Design automation use cases are identified based on the method for identification of design automation use cases. Yet, instead of applying the design automation task templates accounting for the process perspective, the extended templates including the motivational perspective are used. Thus, a list of metrics for estimation of the impact of design automation on design practice is readily available.
2. The metrics associated to the design automation task templates are filtered according to the corporate strategy / interests as well as availability of data. Selected metrics are integrated into the design process models reflecting possible future scenarios of the design process so to link metrics with the actual measurements such as start and end point for measurement of design task lead-times.
3. For the case additional metrics are needed for comprehensive assessment of design automation potential, the method for derivation of metrics is applied. The yielded metrics are added to the design automation task template so as to enable reuse in future scenarios.
4. The final step accounts for qualitative (or quantitative, if possible) estimation of the impact of design automation implementation on metrics. Based on experience as well as informed decisions and data, the impact of design automation implementation on the design performance is estimated. Also, the design process models of current state of practice as well as models of possible future scenarios highlighting changes in the design process structure serve as a means to make informed decisions. Based on availability of data, relative changes to available data sets or qualitative estimations based on the Likert scale (Likert 1932) can be applied to indicate design automation potential. The Likert scale is defined as indicated in Table 8-1.

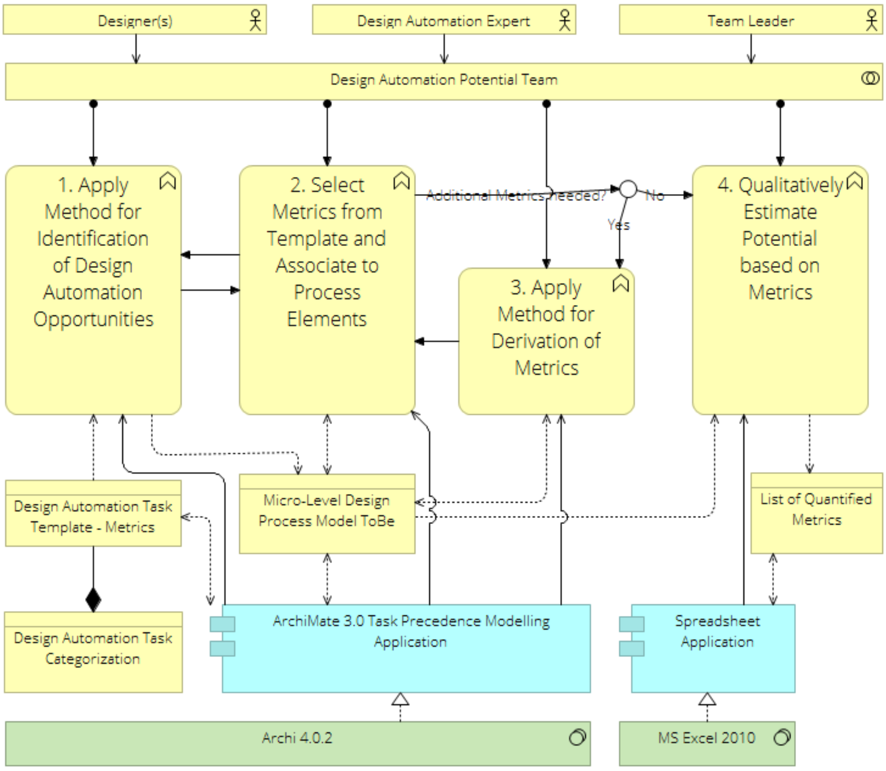


Figure 8-1: Method for identification of design automation potential. The figure highlights the method's main steps including respective data flows (central layer, yellow), the workshops' composition (top layer, yellow) as well as supporting tools and technologies (blue & green).

Table 8-1: Likert scale applied for indication of changes of metrics due to design automation implementation

--	-	0	+	++
Strong decrease of metric value	Decrease of metric value	No impact	Increase of metric value	Strong increase of metric value

Workshop Organization

For conducting the method in industrial practice, multiple workshops are needed due to the iterative nature of the method and its aggregation of the methods for identification of opportunities and (optionally) derivation of metrics. For Steps 2 and 4, a mixed team consisting of at least two designers and the team leader or decision makers is needed to account for different viewpoints for selection and rating of metrics. The collaborative aspects of potential estimation are expected to increase the accuracy of estimations. This is particularly true, since the design process is not only a technical process but also a cognitive and social one (Cross and Clayburn Cross 1995). Hence, designers are required for estimation of potential impacts of design automation on design performance.

8.2 Results

Evaluation of the method for estimation of the impact of design automation on design practice was conducted for the industrial test Case 1 – Structural Analysis and Case 2 – Hydraulic Systems (see Section 4.3). Thereby, the focus is put on the results of the method's Step 4 regarding the rating of the impact of design automation application on metrics. In particular, a comparative study (Blessing and Chakrabarti 2009) is conducted to systematically investigate and address the uncertainties when estimating the impact of design automation on design performance. Thereby, the results attained from the application of the method for identification of design automation opportunities (Section 6.2.2) and the method for derivation of metrics (Section 7.2.2) are reused regarding Steps 1 – 3 and used as a basis for the user study regarding Step 4.

In Section 8.2.1, the research methodology including the addressed research question, the impact diagram and the applied method for conducting the user study are introduced. Following this, Section 8.2.2 presents the results yielded by conducting the study for the industrial cases 1 and 2.

8.2.1 Research Methodology

In order to investigate the impact of uncertainties in estimation of metrics values, the following section presents the applied research methodology. First, the impact diagram is derived to identify the influencing factors that are investigated in this study and the method to conduct the study is detailed. Next, the industrial use case including the background of the workshop participants is introduced. Finally, the experimental setup of the study is presented with a detailed description of the workshop systematic.

8.2.1.1 Impact Diagram & Applied Method

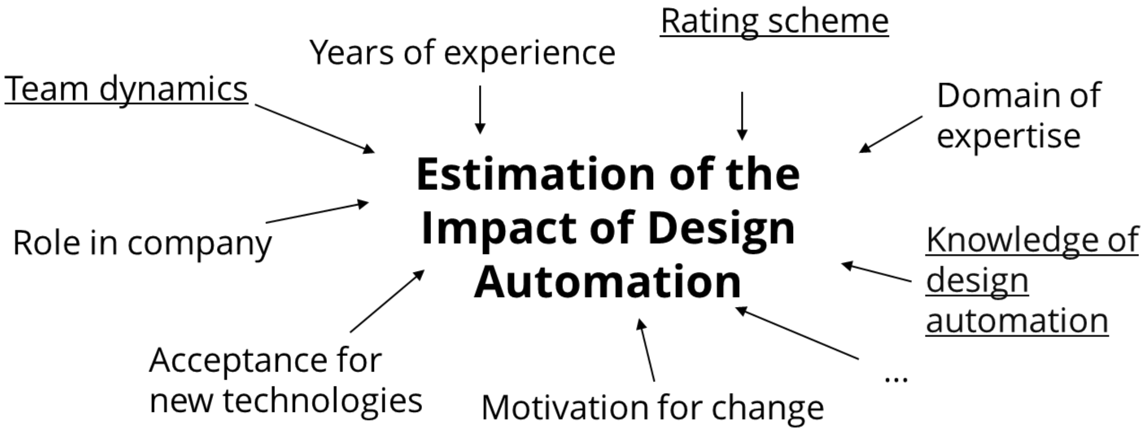


Figure 8-2: Impact diagram for estimation of the impact of design automation highlighting the factors which are assessed in the experiment.

In order to provide an answer to research question five: "How can the impact of design automation on design practice be estimated prior to the implementation?" a more detailed investigation of the estimation of the impact of design automation on design practice is required. In this respect, Figure 8-2 lists the influencing factors for estimation of the impacts of design automation based on assumptions and experience. In particular, the following factors are in focus of this work:

- ◊ Rating scheme: The influence of different rating systems on efficiency and effectiveness of metrics rating needs to be assessed. More specifically, estimation of metrics values based on relative changes of metrics vs. qualitative estimation of changes based on the Likert scale need to be assessed.
- ◊ Team dynamics: The effect of individual rating of metrics vs. team discussions needs to be investigated in order to account for the impacts of team dynamics on metrics rating. Hence, different communication guidelines are established promoting individual or team rating of metrics.
- ◊ Knowledge of design automation: the impact of knowledge about design is of particular relevance, since the majority of designers have little or no knowledge of design automation especially considering the range of design tasks for which design automation can be applied for (Rigger and Vosgien 2018).

In order to assess these influencing factors, a comparative study (Blessing and Chakrabarti 2009) is conducted where the remaining influencing factors such as the participants’ characteristics (year so experience, role etc.) remain unchanged. For investigation of each factor an experimental group and control group needs to be determined. Due to the limited availability of workshop participants for each industrial case, the establishment of control groups for each case was not feasible. Instead, comparison is enabled based on application of different rating schemes and communication guidelines for the cases 1 and 2. Thus, each use case serves as both control and experimental group for investigation of these factors so as to enable qualitative comparison. Regarding the impact of knowledge of design automation on metrics rating, a prototypical implementation of a design automation application tailored to the Case 2 – Hydraulic Systems is applied to demonstrate the capabilities and modes of operation of design automation and thereby improve the participants' knowledge of design automation. The impact of the design automation prototype is investigated with the workshop participants for Case 2 who had no prior knowledge on design automation. Hence, they served as control and experimental group since measurements of metrics rating was conducted before and after showing the design automation prototype. Table 8-2 provides an overview of the applied for the two industrial cases.

Table 8-2: Indication of applied systematics for the comparative study

	Case 1 – Structural Analysis	Case 2 – Hydraulic Systems
Rating scheme	Indication of relative changes	Qualitative rating based on Likert scale
Team dynamics	Team discussion	Individual rating
Experience with design automation	-	**Control group**: Participants prior to demo of design automation prototype **Experimental group**: Participants after demo of design automation prototype

8.2.1.2 Industrial Cases

Regarding *Case 1 – Structural Analysis*, after the derivation of metrics a proof-of-concept implementation of the *spatial product architecture parameter synthesis* task for *automated sizing of sheet metal thicknesses incl. mass optimization* was implemented based on a heuristic method.

For *Case 2 – Hydraulic Systems*, two potential design automation use cases are identified: *product architecture topology synthesis* for the *hydraulic circuit concept generation* and *product architecture parameter synthesis* for *hydraulic component selection for closed circuits*. To illustrate a basic application of design automation as commonly applied in industry (Arora 2004), a design automation prototype for hydraulic component selection for closed circuits is implemented. Therefore, a mixed-integer nonlinear derivative-free algorithm is implemented yielding one cost-optimized solution given the design requirements and the constraints of the system. With respect to hydraulic circuit concept generation, the prototype is implemented based on the design automation method presented by Münzer and Shea (2017). For given boundary conditions, this prototype features automated generation of hydraulic circuits based on first order logic and Boolean satisfiability. Thus, multiple solutions can be generated for given boundary conditions. To illustrate the concept of automated validation of designs, simulations for representative architectures are (manually) created based on existing simulation libraries. As shown by Münzer and Shea (2017), this could potentially also be done automatically. Table 8-3 summarizes the details of implemented design automation prototypes.

Table 8-3: Overview of design automation prototypes including the applied knowledge formalization and reasoning techniques

Design automation prototype	Knowledge Formalization	Reasoning Method
Automated sizing of sheet metal thicknesses incl. mass optimization	Procedural	Heuristic / Enumeration
Hydraulic component selection for closed circuits	Declarative	Mixed-integer nonlinear derivative-free
Hydraulic circuit concept generation	Port-based graphs	First order logic / Boolean Satisfiability

8.2.1.3 Experimental Setup

Regarding *Case 1 – Structural Analysis*, the design automation prototype for automated sizing of sheet metal thicknesses was presented to the designers after the workshop for derivation of metrics. With respect to *Case 2 – Hydraulic Systems*, the workshop for estimation of metrics took place right after the validation of metrics. In contrary to Case 1, the design automation prototypes for product architecture topology and parameter synthesis were prepared prior to the workshop. Whereas the design automation prototype for automated selection of components was already presented, the design automation prototype for product architecture topology synthesis for hydraulic systems had not been introduced to the workshop participants of Case 2 before.

The collaborative workshops for the rating of metrics build on the results of Step 6 (Validate Metrics) of the method for derivation of metrics. The set of validated metrics considered as relevant by practitioners are used as a basis for the workshops. To realize the workshops, a printout of the developed metrics system was provided to each participant. For Case 1, the list of metrics was gradually discussed within the team and impacts of design automation implementation on the metrics were estimated. In Case 2, the metrics related to product architecture topology synthesis were assessed to account for the impact of the design automation prototype for hydraulic circuit concept generation. The metrics in Case 2 were rated on an individual basis for each participant without any interaction among the participants despite sitting in the same room doing the rating simultaneously. After the first rating of metrics for Case 2, the list of rated metrics was collected. Following this, the design automation prototype for automated generation of hydraulic concepts for given boundary conditions is presented. First, the aim of the design automation tool as well as the related task formalization such as inputs and outputs are explained in detail. Next, the different steps of the design automation method are presented as well as some intermediate results, e.g. generated product architecture as well as two representative related simulations. By doing so, the opportunity to generate multiple feasible solutions within a short amount of time is highlighted as well as the requirements and related efforts for design automation task formalization. To assess the impact of the change of knowledge of design automation, a new (blank) copy of the same list of metrics is provided for the participants. Then, metrics were rated anew after the presentation of the design automation prototype, again without any interaction.

Figure 8-3 illustrates the organization of workshops for each industrial case and Table 8-4 provides the overview of workshop participants.

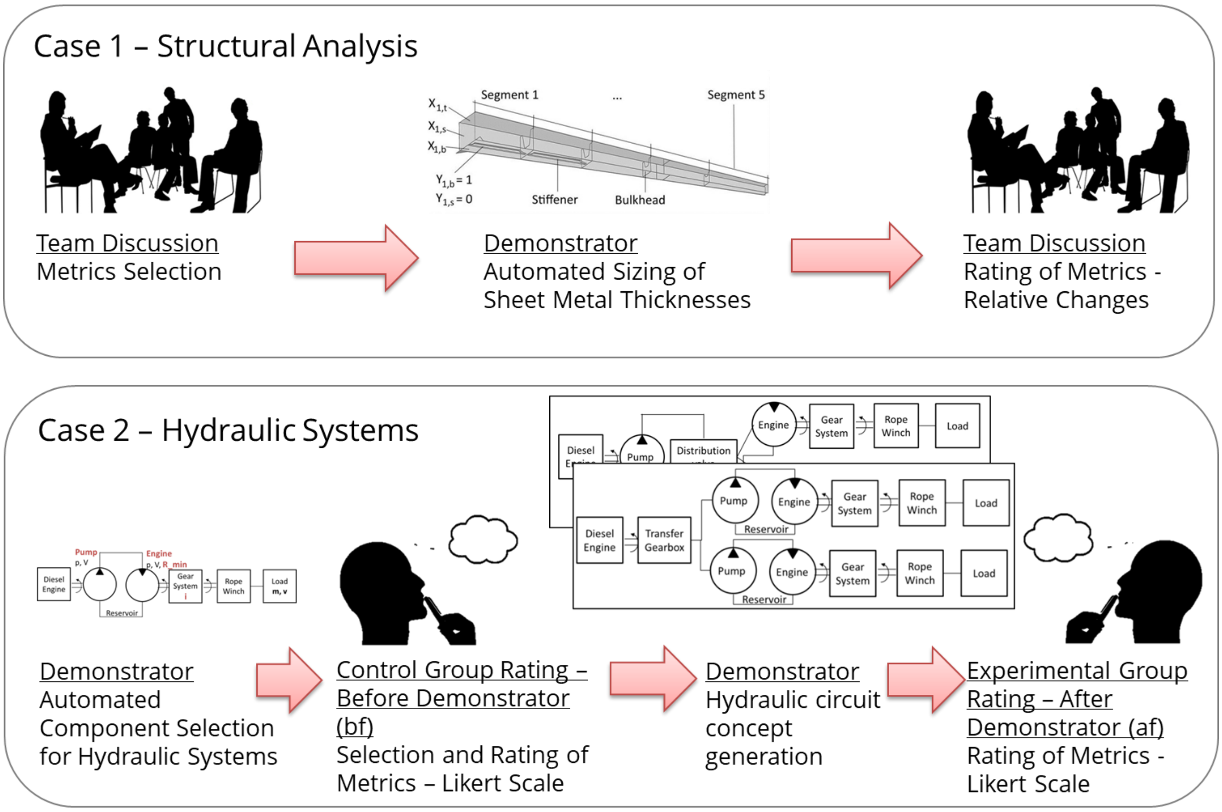


Figure 8-3: Applied systematic for conducting the comparative study

Table 8-4: Workshop participants

Participant	Position	Work experience
CASE 1 – Structural Analysis		
Designer 1-A	Designer, Team leader	4
Designer 1-B	Design Engineer	10
Designer 1-C	Department Manager	12
CASE 2 – Hydraulic Systems		
Designer 2-A	Designer, Team leader	11
Designer 2-B	Designer, Team leader	11
Designer 2-C	Department head	35
Designer 2-D	Designer	4

8.2.2 User Study Results

In the following, first the rating of metrics for Case 1 – Structural Analysis are presented in Table 8-5. The expected relative changes of metrics as yielded in team discussions in the workshop are indicated as well as a brief explanation why these changes are expected. Next, Table 8-6 shows the evaluation of metrics values describing the potential impact of automated hydraulic circuit concept generation on design practice for Case 2 – Hydraulic Circuits. For each metric, the individual rankings of the four participants before and after viewing a design automation prototype are presented based on the Likert Scale. The mean, standard deviation and range of responses are listed for each metric to indicate the difference in evaluations provided by the participants. This information is provided for each metrics twice, once before and once after the demonstration of the prototype. The metrics are layered with green colour if the change in the metrics value is considered beneficial and red for the case that metrics rating is disadvantageous. For the case that the metrics rating did not experience a change larger than 0.25 in mean, the background colour is white. With respect to this, Figure 8-4 shows a histogram highlighting how the rating of metrics varied due to the demonstration. The histogram further differentiates the results for each designer. Since the participants asked for anonymous review of results, the results cannot be mapped to the position and experience of participants. To further illustrate the impact of the prototype on the rating of metrics, Figure 8-5 shows the impact of the metrics rating for a selected set of metrics. In particular, the rating of the metrics that experienced the strongest change of mean values due to introducing the design automation prototype is shown. Further, the rating of the ROI is listed as an example of a metric that did not change.

Table 8-5: Qualitative rating of metrics to indicate design automation potential for automated sizing of sheet metal thicknesses incl. mass optimization

DA Task	Metric	Δ [%]	Reason
Automated sizing of sheet metal thicknesses incl. mass optimization	**Total time for conducting parameter synthesis activity with / w/o DA**	**+ 10**	**expensive computations**
	Number of iterations in process / average number of iterations	**- 100**	**no more iterations with design dep.**
	Cycle time: Lead time per iteration of parameter synthesis	- 10	more robust
	Lead time for sub-activity (simulation) for parameter synthesis	0	not affected
	On-time delivery: Number of Functional components released on time / Total number of Functional Components	0	additional waiting times mitigate gained robustness
	# of rework rate due to defects in input	**- 100**	**no more iterations with design dep.**
	performance manual design vs DA design	**- 10**	**Mass reduction**
	Cost of manufacturing for design with / w/o DA	**- 5**	**Homogenization of thickness distribution**
	# of validated use cases / total # of use cases	**+ 10**	**more time for additional use cases**
	# of evaluated alternative designs per design activity / average # of solution alternatives investigated per activity	**0**	**no changes expected**
	Number of designs generated using DA / # of generated designs	**-**	**DA should account for 80 % of designs**

DA Task	Metric	Δ [%]	Reason
	Computational Expenditures per project / average personal cost per project	**+ 50**	**additional computational expenses**
	Personal Expenditures per project / average personal cost per project	**- 75**	**only adaptations required**

Table 8-6: Qualitative rating of metrics based on Likert scale for Case 2 before (bf) and after (af) presenting a design automation prototype for generation of hydraulic concepts.

	Metric Description		Designer				Mean	Std-Dev	Range
			1	2	3	4			
Hydraulic Circuit Concept Generation	# of concepts discussed in team / # of concepts discussed on average	bf	0	-	-	-	-0.75	0.43	1
		af	+	+	+	++	1.25	0.43	1
	# of stored designs / # of designs created	bf	++	+	+	+	1.25	0.43	1
		af	+	++	+	+	1.25	0.43	1
	# of product failure modes of products delivered within specific duration / # of delivered products in specific duration	bf	0	++	+	0	0.75	0.83	2
		af	+	0	+		0.67	0.47	1
	Ideation Quality: Measures the feasibility of an idea and whether it meets the design requirements	bf	++	0	+	++	1.25	0.83	2
		af	0	+	+	0	0.50	0.50	1
	# of evaluated alternative designs per design activity / average # of solution alternatives investigated per activity	bf	++	0	+	++	1.25	0.83	2
		af	+	0	+	++	1.00	0.71	2
	Is the safety in the design process improved	bf	++	+	++	++	1.75	0.43	1
		af	+	++	+	++	1.50	0.50	1
	Time spent for scanning database for right item / total activity time	bf	+	+	++	+	1.25	0.43	1
		af	-	+	++	+	0.75	1.09	3
	Number of data items relevant to the user's task and accessible / total number of data items relevant to the user's task	bf	0	0	++	+	0.75	0.83	2
		af	+	-	++	++	1.00	1.22	3
	Total expenditures per project / total average cost per project	bf	+	+	++	+	1.25	0.43	1
		af	-	0	0	0	-0.25	0.43	1
	Total time for conducting design task	bf	++	+	++	+	1.50	0.50	1
		af	+	-	+	0	0.25	0.83	2
	On-Time Delivery: Number of FCs released on time / Total number of FCs	bf	+	0	0	0	0.25	0.43	1
		af	+	0	+	+	0.75	0.43	1
	% of time spent for R&D activities / project	bf	-	+	+	+	0.50	0.87	2
		af	-	+	+	0	0.25	0.83	2
	% of relevant ideation activities regarding product, process and other domains by any individual	bf	0	+	0	0	0.25	0.43	1
		af	-	+	+	+	0.50	0.87	2
	Employee satisfaction level	bf	+	0	0	0	0.25	0.43	1
		af	0	-	0	0	-0.25	0.43	1
	Delay times: Mean and deviation, or distribution, of wait times (best)	bf	0	0	+	+	0.50	0.50	1
		af	0	-	+	+	0.25	0.83	2

	Metric Description		Designer				Mean	Std-Dev	Range
			1	2	3	4			
	# of engineering changes for specific design / engineering changes for designs on average	bf	-	+	++	+	0.75	1.09	3
		af	-	+	++	0	0.50	1.12	3
	# of rework rate due to defects in input	bf	0	+	++	0	0.75	0.83	2
		af	-	0	0	+	0.00	0.71	2
	Key performance indicator of product / performance according to specification	bf	0	+	0	+	0.50	0.50	1
		af	0	++	0		1.00	0.71	2
	ROI: yield of time savings / investment cost for implementing DA	bf	+	-	+	0	0.25	0.83	2
		af	+	-	0	+	0.25	0.83	2
	Personal Expenditures per project / average personal cost per project	bf	0	-	0	+	0.00	0.71	2
		af	0	-	0	+	0.00	0.71	2

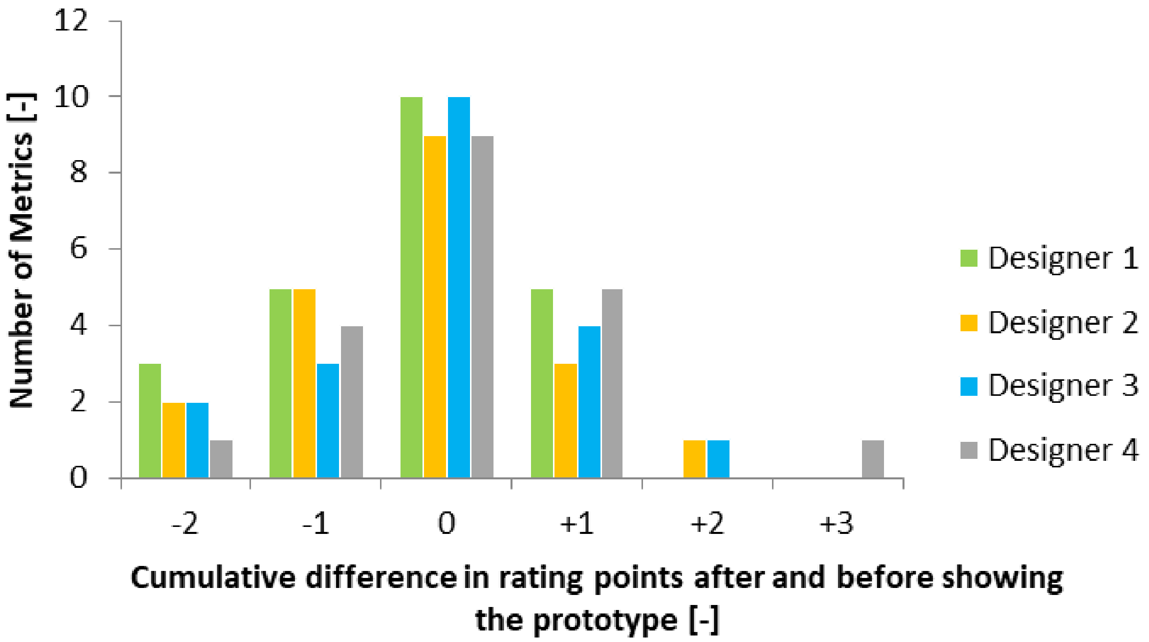


Figure 8-4: Histogram chart indicating the impact of the prototype on rating of metrics for each designer.

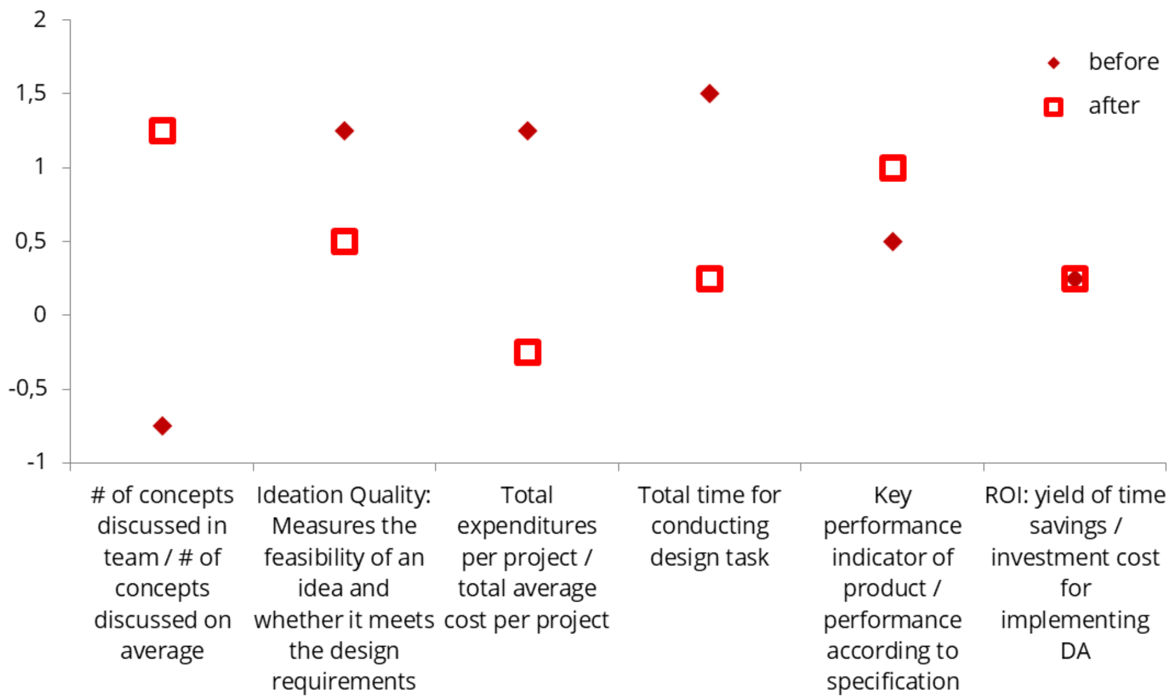


Figure 8-5: Direct comparison of rating of selected metrics that experienced the strongest change due to the demonstration of design automation prototype. ROI is listed as an example of a metric that did not change.

8.3 Discussion

This section discusses the results obtained from application of the method for estimation of the impact of design automation on design practice to two industrial use cases using different systematics for conducting the workshops. First, the different types of ratings applied for Case 1 and 2 are analyzed and their effect on obtained workshops results are discussed. Second, the impact of knowledge of design automation on potential estimation of design automation is assessed based on the attained results for Case 2. Finally, the implications of this research for both design practice and research are highlighted.

8.3.1 Investigation of Rating Scales

With respect to Case 1, a collaborative workshop was conducted to rate the metrics based on numeric indication of the expected change, see Table 8-5. Each metric was gradually discussed in the team based on propositions made by the design automation expert as well as insights gained by a prototypical implementation for spatial product architecture parameter synthesis. It has to be noted that partially the rating of metrics can only be done in conjunction with the design process model indicating the structure of the process, e.g. for metrics related to reduction of iterations or cycles in design. Further, some metrics can only be rated based on expert knowledge indicating the necessity for team discussions and collaboration. For example, the cost reduction is due to homogenization of sheet metal that can be considered a positive side effect attained from applying design automation. Similarly, team leaders / managers are needed within the workshop to account for the strategic aspects of design automation implementation.

For the workshop related to Case 2-Hydraulic Systems, the rating of metrics was conducted on an individual basis without any team discussions. This allows double-checking the reliability of metrics rating based on assessment of the agreement of results among the participants. Further, the Likert scale estimation was applied instead of numeric indication of differences for the sake of simplicity and efficiency of rating. Regarding reliability of rating, it can be seen within Table 8-6 that consistency of metrics among designers is in an acceptable range for the individual rating of designers. This is indicated by the standard deviation as well as the ranges of results that mostly span one or two classes of the Likert scale. The validity of results needs to be considered from the perspective of the expressiveness of the Likert metrics scale. For instance, a metric related to total time for conducting a project can be considered from two perspectives on the Likert scale: first, if the duration for conducting a design task increases more positive ranking sounds intuitive since its numeric value increases. Second, when considering the Likert scale form a motivational perspective, a longer duration for conducting a design task denotes decreased design efficiency. Hence, a negative value needs to be selected on the Likert scale. The first refers to the way the Likert scale was communicated prior to the experiment, i.e. related to numeric changes of metrics. Considering the acceptable range of agreement among practitioners, it can be assumed that the interpretation of the Likert scale was aligned among practitioners.

Finally, when comparing the two different systematics applied for rating of metrics, it can be concluded that the advantages of numerical rating of metrics exceed the ease of usage of the Likert scale. The team discussion guarantee thorough assessment of the metrics from multiple perspectives and communication of results is facilitated based on indication of numeric values / relative changes.

8.3.2 Impact of a Design Automation Prototype

As can be seen within Figure 8-4 and Table 8-6, showing a design automation prototype to designers that are not knowledgeable about design automation impacts the estimation of design automation potential. Figure 8-4 shows that the average rating of metrics slightly decreased. Since the Likert scale refers to the quantitative impact of design automation implementation on a metric, this does not necessarily mean that the overall potential for design automation implementation decreases. For some metrics an increase in the metric (e.g. Key performance indicator of product) is desirable and for some a decrease is beneficial (e.g. total time for conducting a task). Instead, a more realistic estimation of potential impacts of design automation implementation is enabled due to the gained knowledge. For example, the designers stated: “This is a powerful and desirable tool. Yet, this requires some effort to be able to account for the majority of cases. One person needs to permanently work with it“. Based on Figure 8-5, this statement can be validated, since key performance indicators of the product are expected to increase while the expenditures and time for conducting a project decrease. Yet, the ROI remains the same, which is an indicator for the expected efforts for the maintenance of the design automation application.

Further, Figure 8-4 shows that the majority of ratings are not impacted by the design automation prototype. This leads to the assumption that the design automation prototype does not account for all the aspects covered by the metrics. As a result, careful attention needs to be paid not only to the selection of metrics but also to the features shown in the demonstration scenario. For example, total expenditures per project are considered to decrease. Yet, neither the list of metrics nor the demonstration accounts for the calculation costs that are not negligible for that type of automation. To equalize this impact, it is required to present the design automation prototype from different perspectives and that the related metrics are carefully selected. Preferably, the comprehensive list of metrics is gradually discussed based on the design automation prototype and its characteristics to clearly indicate how a metric is impacted by design automation implementation. If not, a design automation prototype biases participants and only partially impacts the rating of related metrics. To further elaborate on the integration of design automation prototypes for communication of the value of design automation, literature from related fields such as marketing research should be considered. For example, studies related to fit-risk that accounts for the customers' concern whether a technology really fits their needs (Parks, Bansal, and Zilberman 2016) is to be considered.

Despite the challenges for implementing a design automation design automation prototype, designers consider it of fundamental support as remarked by one participant: "First it was very abstract. The design automation prototype made clear what we are actually talking about". Therefore, a basic knowledge is needed to be able to make informed decisions about design automation implementation and its benefits.

8.3.3 Implications on Design Practice

The results show that the knowledge on design automation is critical and strongly impacts the evaluation of metrics. Given the fact that designers are knowledgeable about design automation use cases for the early phase, they also see the potential for design automation application as indicated by the metrics rating. This is contradictory to previous findings of a survey that showed that designers do not consider design automation relevant for the early stages of the design process (Rigger and Vosgien 2018). Most likely, this is linked to the lack of knowledge about design automation use cases.

The list of rated metrics provides an overview of potential impacts of design automation on design practice. Yet, the effort for design automation implementation needs to be assessed to finally enable informed decisions whether to implement design automation in practice or not. Existing approaches indicated in Section 2 need to be extended and further elaborated to enable reliable cost estimation of design automation implementation. Also, to further support decision making, methods are needed to summarize the results of metrics rating in a comprehensive manner. The rating of individual metrics needs to be gathered among the different dimensions of design performance, e.g. based on balanced scorecards (Kaplan and Norton 1990). Hence, the method for estimation of the impact of design automation on design practice should be extended to not only account for the estimation of metrics' values but also the rating of relevance of metrics.

8.3.4 Implications on Design Research

The method contributes to design research by presenting a method that features reuse of metrics and relies on design process analysis. Further, different systematics for estimation of implications of design automation are assessed based on user study with designers from industry. The results investigations regarding the impact of knowledge of design automation show that the systematics for demonstrating design automation and its working principles in industrial practice requires further investigation. In particular, different ways to communicate the working principles of design automation should be analyzed, e.g. based on video clips, prototypes or hands-on workshops. User studies are required to investigate various communication media from the perspectives of efficiency and effectiveness. Also, related investigations from marketing research need to be considered. Finally, it has to be noted that generalization or external validity of results cannot be guaranteed. Both, the assessment of systematics for rating of metrics as well as the application of demonstrators requires further investigation. Nevertheless, the findings are the first of its type regarding the comprehensive analysis of estimation of the impact of design automation on design practice.

8.4 Summary

In this section, a method for estimation of the impact of design automation on design practice is presented. The quantification of design automation potential based on metrics builds the basis for informed decisions whether design automation implementation should be pursued or not. The proposed method builds on the methods for identification of design automation use cases and the derivation of metrics. Application of the method to two industrial cases demonstrates that the proposed method enables comprehensive assessment of design automation opportunities. The industrial cases focus on the different systematics to best address uncertainties when estimating the values of metrics. In particular, the impacts of team dynamics, knowledge of design automation and rating scales are investigated. The results indicate that the collaborative character of workshops contributes to accuracy of potential estimation of design automation. A diversity of expert knowledge and positions in the company are required to account for the different viewpoints when rating the metrics. Moreover, the application of numeric scales for indication of changes is required to increase accuracy of estimations and facilitate communication. To address the impact of knowledge of design automation on metrics rating, the rating of metrics before and after demonstrating a design automation prototype is compared. The results indicate the importance of knowledge on design automation to enable reliable potential estimation of design automation.

Based on these findings, the following answer is provided for research question five: "*How can the impact of design automation on design practice be estimated prior to the implementation*?" Knowledge of the working principles of design automation is required to reliably assess the impact of design automation on design practice. Further, collaborative workshops are required to account for the different perspectives of designers on the design process and potential implications. Additionally, the definition of potential future scenarios of the design process and association of selected metrics to the design process enables to contextualize the metrics and the design process. In conjunction with the design automation task templates that account for the specifics of design automation task, the following **first contribution** can be highlighted:

A method that takes into account both the design process and design automation task characteristics and features reuse of metrics for comprehensive estimation of the impact of design automation on design practice.

Regarding the conducted user study the **second contribution** can be acknowledged:

A critical investigation of workshop setups and rating mechanisms is performed to derive requirements for reliable and comprehensive estimation of design automation implications on design practice.

Similarly, the **third contribution** is related to the user study regarding the impact of knowledge of design automation on metrics rating:

The impact of knowledge of design automation is systematically analyzed based on an experimental study and it is shown that demonstration of design automation prototypes impact design automation potential estimation.

The findings are achieved by the application of the method to two industrial use cases. However, they remain limited to findings of qualitative nature due to the limited number of case studies. Future work needs to more comprehensively assess the communication of design automation knowledge as well as methods for evaluation of the design automation opportunities given a set of quantified metrics.

9 Method for Design Automation Task Formalization

The final step of the methodology for design automation task definition focuses on design automation task formalization. In this respect, the formalization of practitioners' knowledge can be considered a major obstacle with respect to design automation implementation in industry. In particular, the black-box perception of design automation due to hard-coded knowledge as well as large efforts for knowledge acquisition and formalization can be considered critical for design automation application in practice, even for relatively simple design automation applications (van der Velden, Bil, and Xu 2012). The first originates from missing means for validation of formalized knowledge resulting in a lack of trust for output generated by design automation and the second leads to high investment costs (Stokes and MOKA Consortium 2001). Hence, a method that enables designers to formalize design automation tasks themselves is needed to increase the understanding of design automation and increase efficiency of task formalization.

Addressing these points, Section 9.1 introduces a method for design automation task formalization that builds upon the findings of sections four to seven. In particular, the design automation task categorization and the findings of the consolidation of the research fields KBE and CDS build the basis of the method. Further, the information obtained from design process analysis is systematically reused. To illustrate the concept and evaluate the method, Section 9.2 presents an industrial user study that focuses on automation of a product architecture parameter synthesis task. Section 9.3 critically reviews the results, analyzes generalizability of the method and discusses its transferability to other types of design automation tasks. Lastly, Section 9.4 presents the concluding remarks and indicates the contributions with respect to answering the research questions six and seven.

9.1 Method

The method presented in this section focuses on its application by designers: the designers whose task is to be automated should be enabled to formalize the corresponding design automation task themselves. As shown in Section 5.2, knowledge formalization is specific to the design automation method as well as the design automation task category. Hence, to enable reuse of knowledge among design automation methods for one specific design automation task, the design automation task formalization needs to be done independent on the design automation method. Further, the design automation task formalization and the mathematical programming according to a specific design automation method need to be separated. Building upon the findings of Sections 2.3 and 5, the usage of SysML is pursued in this work. In particular, SysML semantics are overloaded to account for the specifics of a design automation tasks, e.g. definition of variables for synthesis. Thus, the design automation task formalization can readily be integrated to a model-based systems engineering

(MBSE) process without the need for adaptation of models. Further, the application of SysML syntax enables design automation task formalization in any SysML modeling environment without the need for customization. To guide formalization, design automation task templates are applied, see Figure 5-7. In particular, the data objects identified within the design process are used as a starting point for identification of sources of knowledge, cf. Figure 4-2. The structure of the template not only provides guidance but also supports modularization of the knowledge. Based thereon, the following step-by-step procedure for design automation task formalization can be summarized:

1. The appropriate design automation task template needs to be loaded.
2. The input product knowledge needs to be formalized. Depending on the design automation task templates this involves definition of functions, model libraries, product architectures, geometries, modifications, interconnections and evaluation algorithms.
3. The goals need to be formalized. Depending on the template, functional constraints, geometrical constraints and an objective function need to be defined.

Even though a linear process is described for each template taking into account the specifics of a task, an iterative process is assumed. To enable non-proficient modelers to create the required elements on the diagrams, the templates provide modeling guidelines and instructions on each view. In the following, product architecture topology synthesis (PATS) and product architecture parameter synthesis (PAPS) tasks are used to illustrate the method for design automation task formalization.

Design Automation Task Definition for PATS/PAPS

Figure 5-7 shows the template for PAPS. The package diagram (PKG) is used to provide an overview of the specific types of knowledge required for complete formalization of the design automation task. As indicated in Section 5.3.3, hyperlinks for the displayed packages support navigation among the empty template and the related Block Definition Diagrams (BDD), Internal Block Diagrams (IBD) and Parametric Diagrams (PAR). In the following, the formalization of the input product knowledge and goals for a PATS and PAPS task are detailed as well as a means to review the formalized knowledge. No additional stereotypes are required to the ones already predefined in the SysML specification 1.4 so to rely on established SysML syntax.

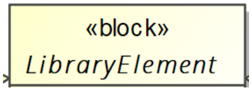
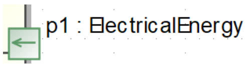
9.1.1.1 Input Product Knowledge Formalization

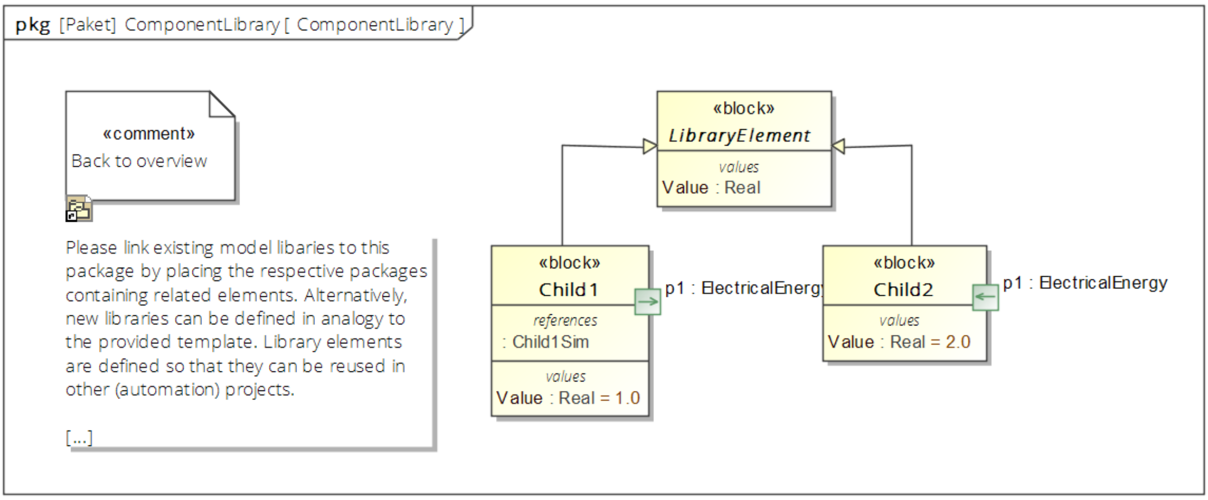
The following subsections detail the proposed systematic for input product knowledge formalization. The headings indicate whether the step applies for PATS, PAPS, or both.

Model Libraries (PATS & PAPS)

First, potentially existing model libraries need to be linked to the "model library" package. Alternatively, a model library can be created from the start based on a provided BDD template. The BDD template serves as a means to illustrate the systematics of model library definition as proposed in (Wölkl 2012; Kruse 2016) and provide a starting point for definition of the model library as illustrated in Figure 9-1. In particular, the concepts of inheritance are applied to define abstraction hierarchies (Dym and Levitt 1991). With this respect, SysML provides *specialization relationships* between blocks as well as the definition of *abstract elements*. For product architecture topology synthesis with focus on energy, signal and mass flow based systems, *directed flow ports* based on the functional basis (Stone and Wood 2000) can be applied to indicate interfaces of components as proposed in (Wölkl 2012). Despite being deprecated, flow ports are still referred to as best practice in systems engineering and will be further considered in future versions (Douglass 2016).

Table 9-1: modeling elements used for definition of component libraries for PATS / PAPS

Model elements	Model type	Symbol	Meaning
Specialization	Specialization Relation		Inheritance of parent block's properties to child block
Abstraction	Block	 «block» *LibraryElement*	The block cannot be instantiated and is used for inheriting properties, only.
Flow Ports	Flow Port	 p1 : ElectricalEnergy	Indication of incoming / outgoing signal / energy / mass flow



Please link existing model libaries to this package by placing the respective packages containing related elements. Alternatively, new libraries can be defined in analogy to the provided template. Library elements are defined so that they can be reused in other (automation) projects.

[...]

Figure 9-1: Generic example of component library illustrating the modeling concepts listed in Table 9-1.

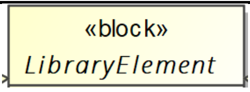
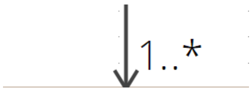
Product Architecture (PAPS)

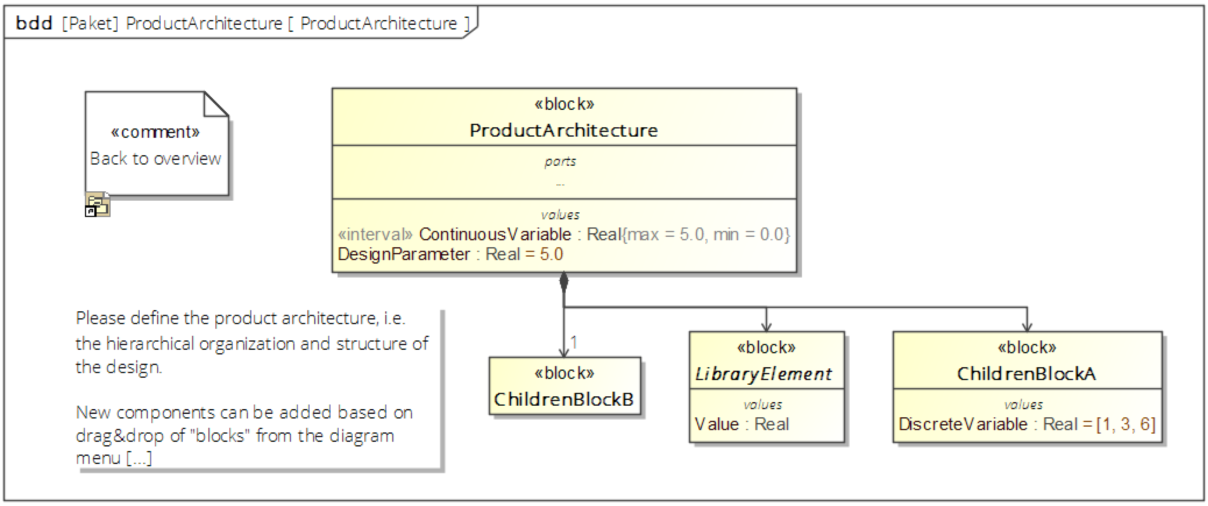
The product architecture can be instantiated based on reuse of SysML blocks from the model libraries, reuse from existing SysML models or by newly defining blocks and composition relations in the empty model. According to the object-oriented paradigm of SysML (Friedenthal, Moore, and Steiner 2014), the hierarchical structure of a system (the product architecture) requires a main block that aggregates the parts (i.e. subsystems / -assemblies / - components). This can be seen in analogy to CAD-design where assemblies are used to aggregate multiple components in one model.

Modifications (PATS & PAPS)

Next to the usage of flow ports to define system boundaries, Table 9-2 lists the elements used for declaration of variables. Value properties of SysML blocks that are not assigned any value or a specific value, respectively, are considered as parameters and are not subject to optimization. The different combinations of variable declaration are combinable, e.g. a library element possibly contains a continuous parameter which is to be determined by the design automation method. For PAPS, modifications can be indicated directly within the product architecture. Within Figure 9-2, the block *LibraryElement* refers to a component that needs to be selected from the component library, *DiscreteVariable* and *ContinuousVariable* refer to variables that need to be determined. For PATS, possible modifications need to be captured by definition of a system boundary based on flow ports, see Figure 9-3.

Table 9-2: modeling elements used on BDD for definition of system boundaries / product architectures for PATS / PAPS

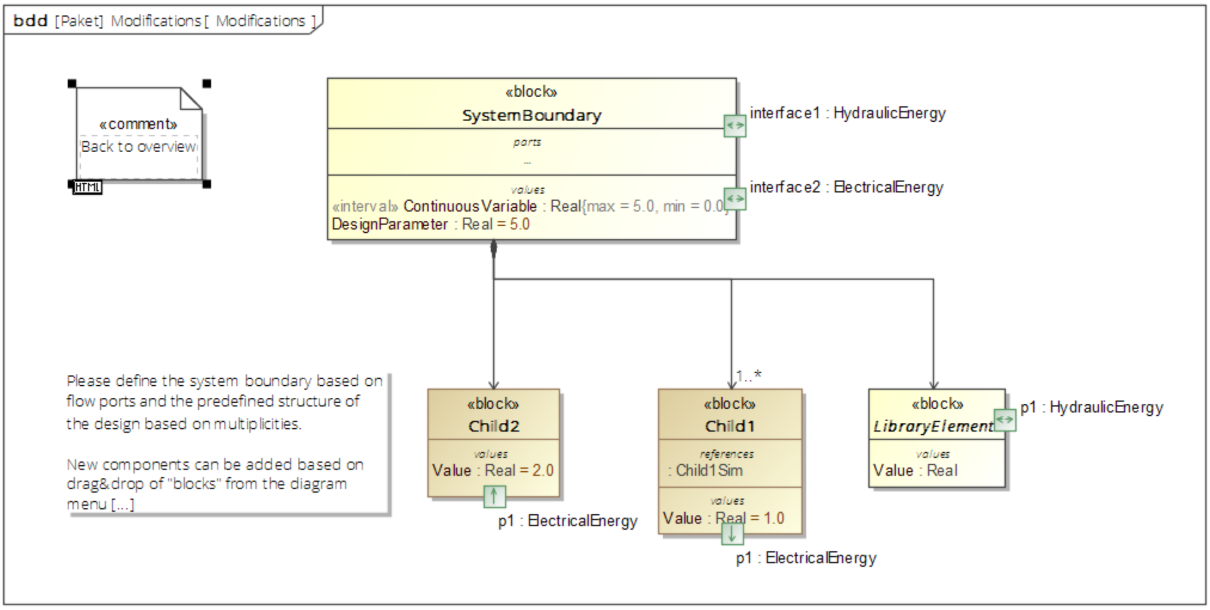
Model element	Model type	Symbol	Meaning	Task
Abstract Block	Block		Indication of a discrete variable for component selection. Works in conjunction with model libraries, only.	PATS / PAPS
Multiplicity	part-of relationship		Indication of parts as discrete variables with multiplicity as lower and upper bound of variable	PATS
Discrete variable range	Value property	[a, b, c, d,…]	Indication of discrete variable (set).	PAPS
Basic interval stereotype	Value property	min = ; max =	indication of continuous variable as well as corresponding variable range	PAPS



Please define the product architecture, i.e. the hierarchical organization and structure of the design.

New components can be added based on drag&drop of "blocks" from the diagram menu [...]

Figure 9-2: BDD template depicting modeling guidelines, the product architecture as well as possible modifications for a PAPS task.



Please define the system boundary based on flow ports and the predefined structure of the design based on multiplicities.

New components can be added based on drag&drop of "blocks" from the diagram menu [...]

Figure 9-3: BDD template depicting modeling guidelines and possible modifications for a PATS task

Interconnections (PATS)

For PATS tasks, the predefined interconnections need to be defined for the case that the topology is partially predefined. In this respect, the flow ports of blocks listed within the system boundary and flow ports of the system boundary can be linked to account for predefined connections see Figure 9-4.

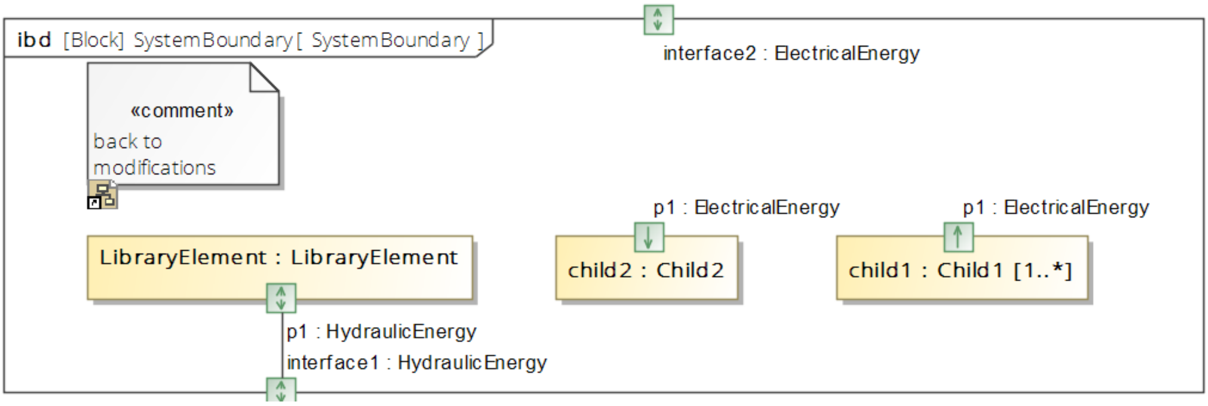


Figure 9-4: Indications of interconnections based on links in IBD.

Evaluation Algorithm (PATS & PAPS)

In order to evaluate functional performance of the generated designs, evaluation algorithms are required. With this respect, a PKG called “Evaluation Algorithm” is provided that contains a textual description of how parametric relations for evaluation of product architectures can be defined. Further, the successively defined PARs can be linked to the PKG to provide an overview of formalized relations. A PAR needs to be defined for each parametric relation between parts, variables and parameters, see Figure 9-5. For the case that external solvers such as an FE-Solver need to be called, SysML “Constraint Blocks” are used to capture the input parameters for model transformation as proposed in (Peak et al. 2007b, 200). Thus, the parametric relations between parts are all captured on the “Evaluation Algorithm” PKG in order to enable fast navigation among the formalized relations. The formalized equations can then be used to evaluate a design. With this respect, a constraint solver can be used to resolve the parametric dependencies among parameters or external simulation capabilities.

Mappings (PATS)

If available, mappings to simulation models based on directed associations can be defined for evaluation of PATS, see Figure 9-6. Previous work has shown how the SysML can be extended to account for the specifics of Modelica (Gauthier et al. 2015) or Amesim (Kruse 2016) simulation environments so that simulation models can be automatically created.

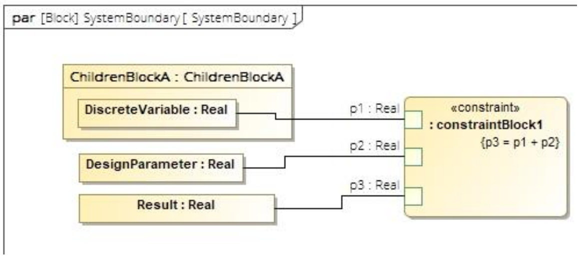


Figure 9-5: Formalization of parametric constraints based on constraint blocks.

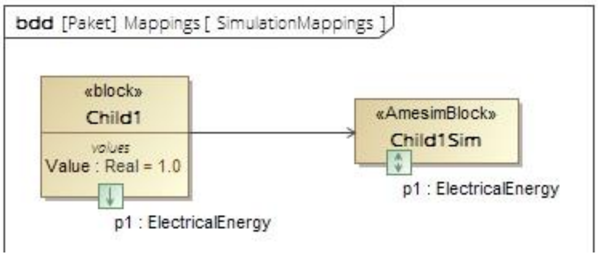


Figure 9-6: mapping of simulation models based on directed associations.

9.1.1.2 Goals Formalization

Considering the formalization of goals, functional constraints need to be defined to assess conformance of generated designs to the specifications. The definition of functional constraints follows the same procedure as described in the previous step for definition of parametric relations: constraint blocks are defined to indicate parametric dependency by means of constraints (<, >, <=, >=, =). The corresponding PARs are then linked to the “Functional Constraints” PKG for organization of formalized constraints.

The last step encompasses the definition of the objective function to indicate the key performance indicator of the generated designs, e.g. cost. To clearly indicate the objective function and distinguish it from constraints and parametric relations, the <<ObjectiveFunction>> stereotype which is predefined in SysML is applied to the constraint block. The value properties corresponding to the key performance indicators need to be highlighted based on the SysML stereotype <<moe>> (measure of effectiveness).

9.1.1.3 Review of formalized knowledge

For assessment of interdependencies of parameters, relation maps can be used for analysis of already defined parametric relations as indicated in Figure 9-7.

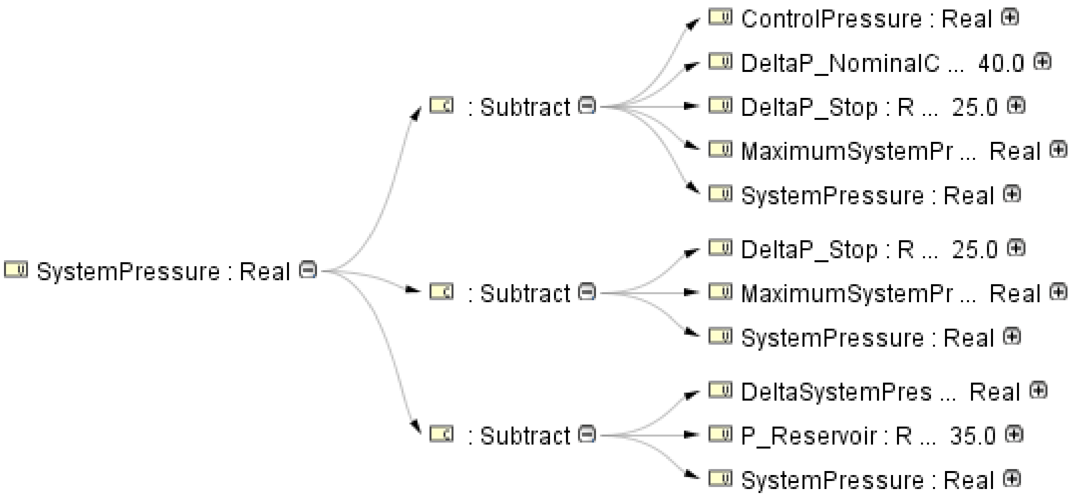


Figure 9-7: Relationship diagram indicating dependencies among value properties and constraint blocks

However, to assess semantic completeness of the formalization, the transformation to the computable formalization needs to be conducted in order to attain feedback on completeness and solvability of the formalized task. This is itself an iterative process as indicated in Arora (2004).

9.2 Results

For evaluation of the method for design automation task formalization, a user study was conducted based on industrial Case 2 – Hydraulic Systems. In the following, Section 9.2.1 first introduces the research methodology including the addressed research questions, related research hypotheses and the applied method for conducting the user study. Following this, Section 9.2.2 presents the corresponding results.

9.2.1 Research Methodology

In the following, first, the research hypotheses are introduced in 9.2.1.1. Next, the impact diagram is derived to identify the influencing factors that are investigated in this study and the method to conduct the experiments is detailed. Finally, the details of the user study are presented in Section 9.2.1.3. In particular, the experimental setting, the evaluation procedure and the tasks are introduced.

9.2.1.1 Research Hypotheses

Regarding research question six “*In what aspects does the usage of a graphical modeling language support designers to formalize design automation task themselves?*”, the following hypotheses are formulated:

1. SysML based graphical formalization supports efficiency of design automation task definition.

2. Reuse of knowledge from model libraries supports efficiency and effectiveness of design automation task definition.
3. SysML-based design automation task formalization enables design automation method independent formalization of a design automation task.

For research question seven “*How can the completeness of a task be assessed in order to support designers when formalizing a design automation task?*”, the following hypothesis is posed:

4. Modularization of the knowledge base with respect to types of input knowledge and goals enables qualitative assessment of completeness.

9.2.1.2 Impact Diagram & Applied Method

Design automation task formalization conducted by designers can be considered a usability problem. The key elements of interest are efficiency and effectiveness for formalization of a design automation task. In particular, focus is put on whether a designer is able to formalize a design automation task and to what extent to indicate effectiveness. Further, the effort needed to formalize a task is analyzed to account for the efficiency. Whereas the user study aims at assessment of the usability of the method, the study does not aim at the assessment of the modeling tool’s low-level usability, e.g. “how to create a block”. Figure 9-8 highlights that multiple influencing factors need to be taken into account when setting up a user study, ranging from motivational aspects to more skills oriented aspects such as knowledge on SysML or design automation methods. Hence, for quantitative evaluation and statistical significance of results, large sample sizes / number of participants are required to equalize the impact of different backgrounds (Robinson 2016). Addressing this issue, the practice of “discount usability engineering” has been established in the early 1990s (Nielsen 1993). It aims at identification of major usability issues and propagates small qualitative user studies consisting of three to five users, well defined scenarios and a simplified think-aloud method (Nielsen 1994). Regarding the latter, instead of video-taping the user study, the experiment is conducted for each participant individually with an observer taking notes during the session with respect to key characteristics, e.g. number of support calls, time, or general statements by participants. The application of the method yields at a “70% chance of finding 80% of the usability problems in a given product” (Faulkner 2003).

Due to the conceptual character of this work and the limited availability of use case scenarios, the “discount usability method” was selected as the method of choice to conduct the user study. The experimental factor for investigation addressed in this user study is the design support provided by the SysML based method for design automation task formalization. To address external validity of the result, participants who have basic programming knowledge but don’t consider themselves capable of formalizing a design automation task themselves are used as participants. Hence, the necessity of a control group with respect to non-proficient programmers becomes obsolete and the goal of the method to enable non-programmers to formalize a design automation task is assessed.

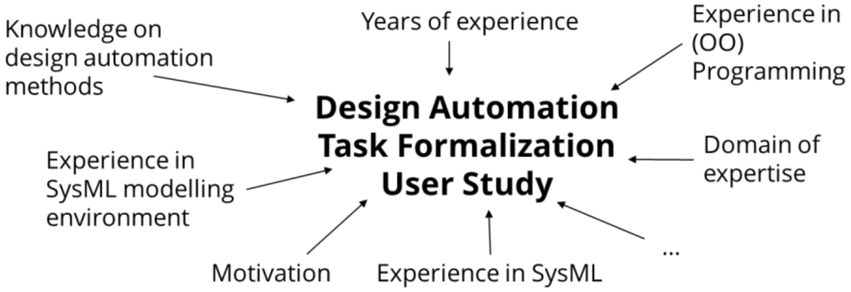


Figure 9-8: Potential influencing factors for evaluation of the SysML-based method for design automation task formalization

9.2.1.3 Experimental Setup - Hydraulic System Component Selection

In this section, first the experimental settings are introduced. Following this, the applied evaluation scheme is presented and the test cases used for conducting the experiment are detailed.

Experimental Setting

The user study was conducted with 3 out of 4 participants of Case 2 – Hydraulic Systems as listed in Table 9-3. All participants have a degree from a university of applied sciences (DI (FH), MSc, respectively) and had a basic introduction to programming at university. None of them have knowledge on systems engineering, MBSE, SysML or the modeling application MagicDraw 18.0.1 that is used in the study. The user study was conducted on an individual basis with each participant and the thesis author. The thesis author remained passive during the study and took notes unless the participants asked for support.

Table 9-3: User study participants

Participant	Position	Work experience	Programming	SysML
CASE 2 – Hydraulic Systems				
Designer 2-A	Designer, Team leader	11	Ok	No
Designer 2-B	Designer, Team leader	11	Little	No
Designer 2-D	Designer	4	Ok	No

For the user study, a 1-hour introduction was provided by the thesis authors for all three user study participants in one session. First, the motivation for the model-based approach is provided: The advantages of graphical formalization as opposed to code-based (mathematical) formalization were pointed out as well as the necessity to enable reuse of knowledge and modularization of the knowledge to support maintenance.

Next, the basics of the method for design automation task formalization are introduced with respect to the systematics of the templates and the respective step-by-step procedure. Further, the basic elements of SysML and the overloaded syntax for PAPS and PATS definition are introduced. Following this, the individual sessions with the participants were conducted.

Evaluation Procedure

To assess applicability of the SysML based method for design automation task formalization from different perspectives, two scenarios were set up: Once, regarding the adaptation of an existing design automation task formalization for PAPS, and second, with respect to design automation task definition of a simple PATS task from the empty template where the definition of interconnections can be omitted. These are further referred to as scenario 1 and scenario 2, respectively. Scenario 1 corresponds to the type of adaptive design automation implementation and scenario 2 to original design as referred to in 6.3. According to the step-by-step procedure described in Section 9.1, each scenario is split up in multiple sub-activities for design automation task formalization according to the modularization of the design automation task template, e.g. “define model library”, “define product architecture”, “define variables” etc. see Section 9.1. Thus, the success can be recorded for each sub-task individually. For setting up the scenarios, particular focus was put on definition of tasks that are neither “too easy or too difficult to solve” and address the motivation of the participants (Dinar, Shah, et al. 2015).

Regarding measurement for evaluation of results, the following aspects are taken into account:

- ◊ the time to conduct a sub-activity,
- ◊ the number and type of errors to conduct a sub-activity,
- ◊ the number of support-calls for completing a sub-activity,
- ◊ the workload perception after each task based on the NASA-TLX test (Hart and Staveland 1988).

Regarding the support calls, the observer who participates in the study provides the desired support if necessary. As stated above, it is not the objective to assess the modeling tool used. Therefore, support-calls addressing modeling issues are not recorded.

Test Cases

In order to increase motivation of designers to participate in the user study, a scenario closely related to a practical problem encountered for a design automation prototype was selected. In particular, the following scenarios were defined to assess efficiency and effectiveness of the approach:

1. “The prototypical implementation of the PAPS for automated component selection currently considers functional performance optimization for a simple closed loop hydraulic circuit. For a fixed hydraulic layout, components and control parameters are defined so that a combination of maximum velocity of an empty hook and maximum full load is achieved. However, the cost of the system is currently not taken into account. To do so, the design automation task needs to be adapted as follows:
    a. The objective functions needs to account for minimization of costs of the engines and the pump.
    b. Values of the specifications with respect to minimum empty hook velocity and minimum full load capacity need to be put as constraints.

    To accomplish the task, the following actions are needed:
    - ◊ The costs need to be added as properties to the component library. Further, components that should not be considered for the optimization need to be removed from the library.
    - ◊ The objective function needs to be adapted to account for the costs
    - ◊ The constraints for the minimum empty hook velocity and full load capacity need to be defined”
2. “For maintenance of a hydraulic system on site, a backpack with tools needs to be packed. Each tool has a specific weight [kg] and enables to do services that generate revenue of a specific value [€]. However, there is a limited amount of number for each tool that can be packed and the backpack must not exceed a maximum weight.

    The goal is to pack the backpack so that maximum revenue can be generated when doing the maintenance. Table 9-4 lists the tools, corresponding weights, values and maximum number that can be packed.

The evaluation schema to assess the user performance for the scenarios is presented in Table 9-5 and Table 9-6.

Table 9-4: specification of tools for user-study scenario 2

Tool	Weight [kg]	Value [€]	Max. Number
Screwdriver	0.5	10	5
Disc grinder small	2	50	1
Disc grinder large	4	100	1
Soldering iron	0.25	25	1
Wrench	0.75	12	5

Table 9-5: assessment schema for user study scenario 1. For each subtask 1 point can be achieved

Main task	Subtask	Assessed concept
Adapt component library	Define costs in library	Inheritance in model libraries
	Cost defined for 4 blocks	Explicit redefinition of properties
	Remove currently not used blocks from library	Delete from model vs. delete from diagram
Adjust objective function	Delete objective function	Delete from model vs. delete from diagram
	Define new constraint	Constraint implementation
	Assign stereotype to objective function	Flags based on stereotypes
	Load properties of engine and pump	Link properties to constraint
Define constraints	Create PAR	How to define diagrams
	Load properties of related elements	Object oriented organization of constraints
	Define constraint block - Properties	Link properties to constraint
	Define constraint block - Constraint	Link properties to constraint
	Define constraint block - Binding connectors	Link properties to constraint

Table 9-6: assessment schema for user study scenario 2. For each subtask 1 point can be achieved

Main task	Subtask	Assessed concept
Define library elements	Apply concept of specialization	Concept of inheritance
	Define blocks	Identify relevant parts
	Define properties	Identify relevant properties
	Define type of properties (int, real, string etc.)	Programming principles
Define system boundary	Define system block	System boundary definition
	Define system properties	Identify relevant properties
	Define part-of relations based on library elements	Model reuse
	Define multiplicities	Definition of variables
Define evaluation algorithm	Define constraint block for weight calculation	Performance evaluation based on equations
Define functional constraint	Define constraint block for maximum weight	Formulation of constraints
Define objective function	Apply stereotype “moe”	Identify objective
	Define objective function	identify and formalize relevant equation
	Apply stereotype “objective function”	Flags based on stereotypes

9.2.2 User Study Results

To indicate efficiency and effectiveness of the method, Figure 9-9 and Figure 9-10 show the achieved points and the time taken to complete a task for each participant for scenario 1 and 2, respectively. Figure 9-11 shows the results of the TLX test to indicate the perceived workload of the participants for each scenario. More elaborate statistical evaluation is omitted due to the limited number of participants. The final SysML models are shown in Appendix A.6.

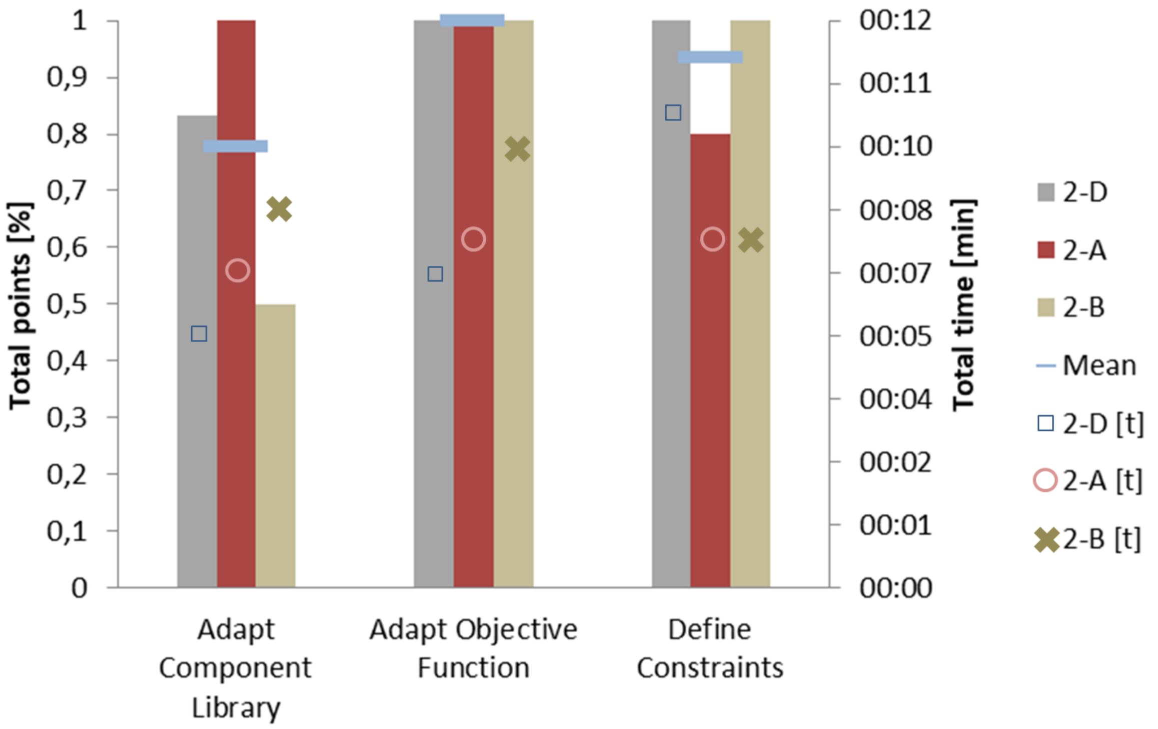


Figure 9-9: number of achieved points, mean of achieved points and total time taken for each subtask for scenario 1 per participant

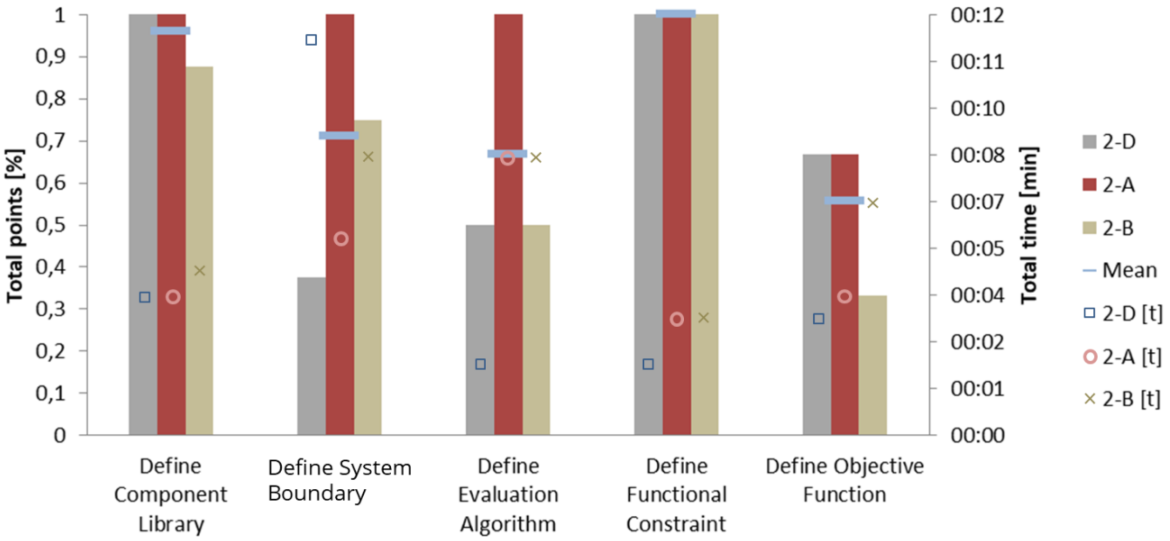


Figure 9-10: number of achieved points, mean of achieved points and total time taken for each subtask for scenario 2 per participant

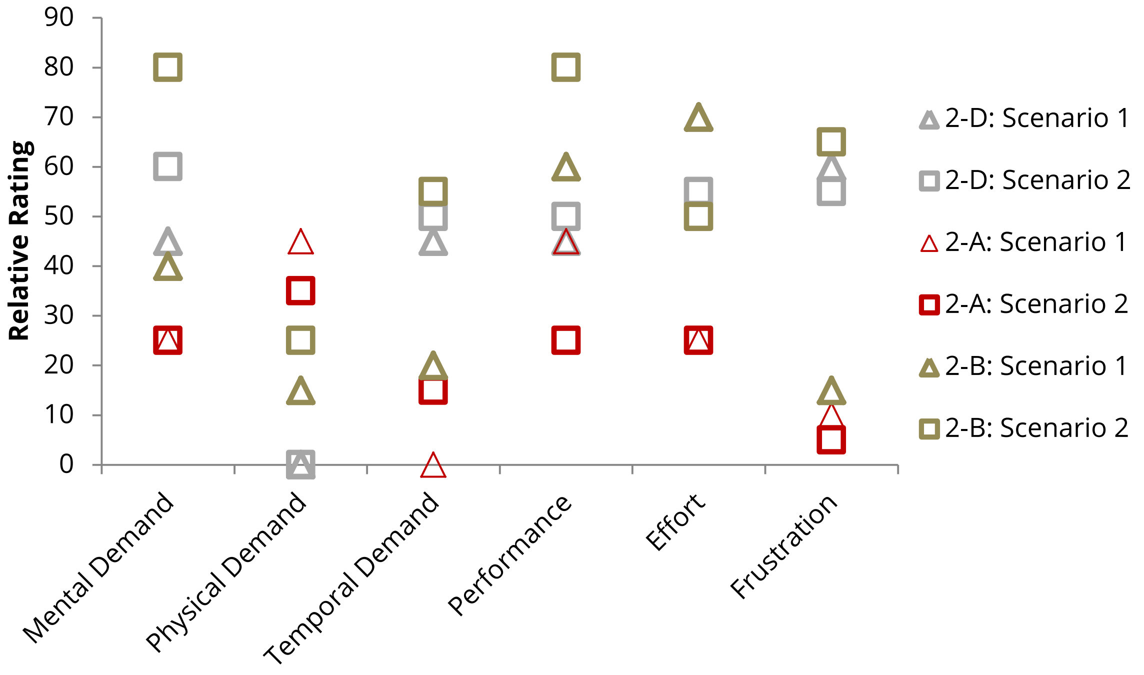


Figure 9-11: Result of TLX-Test measured after completing Scenario 1 and 2 indicated for each participant

9.3 Discussion

This section discusses the results obtained from the user study that aims at evaluation of the method for design automation task formalization. First, the results of the user study are assessed from different viewpoints to indicate the impact of the results on design practice. In particular, efficiency, effectiveness, usability and user satisfaction are analyzed and put into context with the research hypotheses presented in Section 9.2.1.1. Following this, the proposed method is compared to other methods that aim at design automation task formalization by engineers to indicate how this work contributes to state-of-the-art. Finally, the findings and characteristics of the user study with respect to validity and reliability are assessed to highlights the implications of this work for design research.

9.3.1 Indications of User Study and Implications on Design Practice

The user study was conducted with non-proficient programmers without experience in object-oriented programming and modeling. Nevertheless, the majority of tasks were successfully accomplished and the required time of less than half an hour per task indicates high efficiency of the approach. Further, learning effects were observed, e.g. for participant 2-B for the tasks regarding component libraries. Yet, the results of scenario 2 regarding the definition of the system boundary and evaluation algorithm indicate that participants faced difficulties for tasks where task formalization from the empty model is necessary. In fact, analogies appear to be important to the designers to provide guidance as indicated by the observation that one participant could only continue the definition of the product architecture after the analogies to the previously solved task of scenario 1 were explicitly indicated. This indicates that participants prefer to rely on analogies, reuse and adaptations of already existing models. Considering that the integration of validated knowledge has been shown to increase model quality (Kruse 2016; Kerzhner 2012), research hypothesis 2 "Reuse of knowledge from model libraries supports efficiency and effectiveness of design automation task definition." can be strengthened. Also, the TLX test results (see Figure 9-11) show that performance is rated high and the temporal demand is low. Hence, hypothesis 1 "SysML based graphical formalization supports efficiency of design automation task definition." can be strengthened also based on the subjective perception of the participants.

Regarding the organization of the formalized knowledge, the concept of modularizing applied by the task templates and the related navigation within the model was appreciated by the participants. In particular, the graphical modeling for identification of interdependencies was remarked as more transparent, thus more traceable when compared to code. Hence, this method addresses the need for a modularized, accessible and communicable knowledge base (Stjepandić et al. 2015). Also, it enables qualitative evaluation of the completeness of the task formalization with respect to already formalized components and interdependencies. Consequently, research hypothesis 4 "Modularization of the knowledge base with respect to types of input knowledge and goals enables qualitative assessment of completeness" can be strengthened. However, further investigations regarding scalability need to be conducted with more complex design automation tasks.

The results of the TLX test depicted in Figure 9-11 show that the mental demand, effort and frustration correlate with the complexity of the scenarios. Hence, higher values were recorded for scenario 2 than for scenario 1. Potentially, these values could be further decreased when the intuitiveness of the tool support increases and additional means to guide design automation task formalization are applied. For example, workflow patterns and wizard could be implemented to make the formalization more interactive and responsive. Further, the object constraint language (OCL) could be applied to further restrict modeling and avoid semantic errors in models. Yet, the application of OCL constraints makes the integration of the method to existing MBSE environments more difficult since it potentially causes conflicts with existing modeling conventions which is contradictory to the idea of using SysML without any extensions. To further increase usability, also the used SysML modeling environment could be simplified since only a small part of SysML is required for formalization of the design automation tasks.

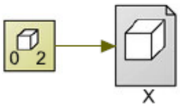
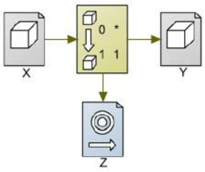
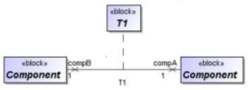
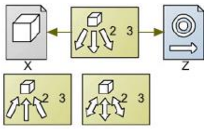
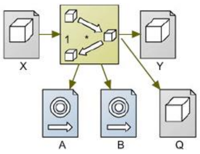
When asked for the overall satisfaction on a scale 0-100, all participants provided answers with values above 75. In particular, they pointed out the reduced abstraction level of graphical modeling for task formalization, however, urged that some concepts require learning of the syntax. Yet, they highlight the benefit of the approach of making the formalized knowledge accessible to them and to support communication. They stated that if the knowledge is not accessible to them, they don't consider design automation a useful approach. Taking into account that the multidisciplinary nature and complexity of design tasks often require designers of multiple disciplines to be involved for design automation task formalization, communication is crucial. Yet, the collaborative aspects of design automation task formalization have not yet been investigated and are subject to future work.

9.3.2 Comparison to related approaches

Only few methods exist that particularly aim at enabling designers to formalize design automation tasks themselves. Interestingly, literature with this focus is mostly related to product architecture topology synthesis. Hence, a comparison of three different methods for synthesis of product architecture topologies is provided here. The first approach considered (Wyatt et al. 2012) originates from an effort to support product architecture design based on constraint networks. It aims at an easy to use formalization of a computational problem. However, the generation of networks does not include performance evaluation, hence, is limited to the generation of product architecture topologies without the possibility to assess them based on parameters. Kerzhner (2012) focuses on declarative formalization of mixed integer linear optimization problems within SysML. The approach features definition of abstract components that refer to the degrees of freedom in the architecture and account for the variability in topologies. The approach relies on model libraries that contain the necessary knowledge for performance assessment as well as predefined interconnections. Finally, Münzer and Shea (2017) present a port-based approach for automated synthesis of product architecture topologies and the related parameters. In order to provide a high level of genericity, the approach relies on labeled ports. This makes the explicit definition of (possible) connections between components unnecessary. Whereas the introduced approaches focus on formalization of PATS according to a specific reasoning methods (constraint solver (Wyatt et al. 2012), MLP optimization (Kerzhner 2012), SAT solver (Münzer and Shea 2017)), the focus in this work is put on providing a means to formalize the design automation task in an intuitive, structured manner and independent on the design automation method.

Table 9-7 provides a comparison of the methods and puts them into context with the method proposed in this work. The table shows that the proposed method reuses parts of the previously published methods and systematically extends these to achieve the level of expressiveness required for each design automation method. Hence, research hypothesis 3, "SysML-based design automation task formalization enables design automation method independent formalization of a design automation task." can be confirmed for the two investigated design automation tasks that focus on product architecture parameter and topology synthesis.

Table 9-7: Comparison of formalizations for PATS proposed in literature and relation to the approach proposed in this work.

Type of constraint / feature	(Wyatt et al. 2012)	(Kerzhner 2012)	(Münzer and Shea 2017)	This work
Level of predefined architecture	All possible connections need to be indicated via constraints definition	All interconnections need to be explicitly indicated. Abstract components indicate variability in topology	System boundary needs to be defined	System boundary and fixed links need to be defined.
Component Number Constraint		 via SysML block multiplicity	in code	Via SysML block multiplicity
Possible connections between components		-	indicated by ports	indicated by ports
Fixed connection of components	Can be done by setting the Multiplicity to 1..1	 via SysML ports and connection templates	Reflected in system boundary	Predefined in system boundary based on binding connectors between ports
Fan out Constraint		components are predefined and connected via SysML Interface. No variability possible	multiple components of type x with different number of ports of type Z	multiple components of type x with different number of ports of type Z
Indirect connection Constraint		-	Intermediate component Q needs to be defined with corresponding in/outputs	Intermediate component Q needs to be defined with corresponding in/outputs
Evaluation algorithm	not considered	based on connection templates and stereotypes	Mapping to simulation models	based on PARs for parametric relations or mapping to simulation models

Type of constraint / feature	(Wyatt et al. 2012)	(Kerzhner 2012)	(Münzer and Shea 2017)	This work
Parametric constraints	not considered	Constraints are stored in model library	in code	based on PARs
Component selection	Abstract components can be used to put a constraint on multiple objects. Abstract does not relate to the concept of inheritance but are used as placeholders.	Possible connections between all components need to be defined. Connections can be classified as optional constraints.	Based on parameters. No component selection possible.	Constraints are defined via ports and abstract components be automatically derived based on network analysis.

9.3.3 Implications on Design Research

The proposed method contributes to design research by consolidating research on different design automation methods for PATS/PAPS as indicated in Table 9-7. The analysis of used syntax and definition of a high level syntax that accounts for the different aspects of different formalizations can be used as a basis for development of new design automation methods. However, the scope of the present work is limited to product architecture parameter and topology synthesis tasks and should be extended for the remainder of the categories of the design automation task categorization. Based on this, a common language for design automation task formalization can be established.

Regarding the evaluation of the method, this work provides a first step towards industrial evaluation of design automation task formalization. Existing user studies are rare and focus on participants from academia, e.g. (Wyatt et al. 2012). The presented user study features internal validity of results since all experiments are conducted based on the same conditions and with the same interviewer. The measurement and rating of results is normalized based on a predefined selection of aspects that are rated and a fixed rating scheme. Solely, qualitative findings are recorded in addition to the rating of topics based on notes taking. On the other hand, the results obtained by the user study cannot claim full external validity. Yet, since applying a well-established method for conducting the experiments, and selection of both domain specific and general scenarios, it can be claimed that the major strengths and deficiencies of the method for design automation task formalization could be identified. Considering the limited availability of resources for conducting the experiments in industrial practice, the approach of discount usability bears potential to support research on design automation task formalization or design support tools in general. In conjunction with the iterative approaches proposed for research on design tools, e.g. (Jensen 1999; Blessing and Chakrabarti 2009), this method facilitates close collaboration with industry based on dramatic reduction of required resources.

9.4 Summary

In this section a method to enable designers to formalize a design automation task is presented. The method is illustrated for product architecture topology and parameter synthesis tasks. An industrial user study is conducted for evaluation of the method as well as a detailed comparison with different formalizations of design automation methods. The application of an established method from user experience research enables to derive qualitative conclusions despite the limited number of participants. This enables to provide an answer to research question six: "*In what aspects does the usage of a graphical modeling language support designers to formalize design automation task themselves?*" In fact, the user study shows promising results indicating that the SysML based modeling of a design automation task makes the task formalization more accessible to designers. In particular, the level of abstraction is reduced based on the application of modeling techniques to enable designers without specific coding knowledge to formalize a design automation task. Object-oriented features such as inheritance can be used to enable reuse of knowledge. Further, the modularization of the required product knowledge provides guidance for formalization and also enables qualitative assessment of the completeness of the design automation task formalization. The latter also provides an answer to research question seven: "*How can the completeness of a task be assessed in order to support designers when formalizing a design automation task?*" Yet, from a semantic point of view, the task formalization can only be fully evaluated when assessing the results yielded by a design automation method application. In this respect, future work should address the automated transformation of SysML based design automation task formalization to different solvers. In particular, the automated tuning of design automation methods according to the characteristics of the task formalization needs to be investigated. Considering these findings and usage of a standardized and established modeling language without the need for extensions, the following **first contribution** can be highlighted:

A method is presented that fully relies on a standardized language and features graphical modeling, reuse and modularization of knowledge.

However, the results presented here are limited to two specific design automation tasks. To generalize the results, more different design automation tasks need to be investigated and corresponding user studies need to be conducted. Still, the presented systematic for comparing different approaches to consolidate efforts for specific design automation methods can be considered as the **second contribution**:

The consolidation of different design automation methods for product architecture topology synthesis is performed to derive a high-level syntax that enables automated transformation of the SysML model to executable code according to the design automation method.

Finally, the work presented here evaluates the method in an industrial context. It is shown how an established method from the field of user experience can be applied to yield significant insight on usability of the method despite the limited access to participants. Hence, the attained results can be acknowledged as a **third contribution**:

An industrial user study is presented that evaluates design automation task formalization for designers and assesses usability of the proposed approach in an industrial context.

10 Discussion

The aim of this section is to discuss the methodology for design automation task definition from multiple perspectives: first, the research methodology pursued in this work is assessed. Following this, the implications to design practice and design research are discussed based on the results from evaluation. Based thereon, the contributions achieved in this work are highlighted and the answer to the global research question is provided: "What are the key factors to increase design automation application in industry?" Finally, the section closes with indication of limitations and outlining the future work on design automation task definition.

10.1 Applied Research Methodology

In this work, the descriptive-prescriptive-descriptive systematic as proposed in the DRM was pursued. The detailed literature review conducted within the first descriptive study allowed to elaborate on the research gaps, establish a design automation metrics system as well as derive the design automation task categorization that is based on a comprehensive analysis of state-of-the-art design automation methods stemming from KBE and CDS research fields. Within the prescriptive study, the four step methodology for design automation task definition was successively developed. The four steps were defined in the prescriptive study and evaluated with three industrial cases within the second descriptive study. Yet, an iterative research process was pursued since the industrial case studies were not conducted in parallel. This allowed readily integrating the findings of the industrial evaluation to the development of the remaining methods. For example, the evaluation for Step 1 (identification of design automation use cases) was conducted before developing the details of the methods to support the Steps 2 – 4.

The evaluation was conducted with industrial cases in the mechanical engineering domain considering different companies and products. The methodology was evaluated by conducting each step of the methodology to the industrial Case 2 – Hydraulic Systems and parts of the methodology were applied to Case 1 – Structural Analysis and Case 3 – Hydraulic Aggregates. For evaluation of each step, focus was put on identification of appropriate methods for conducting evaluation to attain reliable and valid results despite the limited number of use cases and participants. Case 1 – Structural Analysis and Case 2 – Hydraulic Systems focus on design of sub-systems of large-scale designs, Case 3 – Hydraulic Aggregates can be considered as design of a product. All the industrial cases were conducted within multinational companies and for design processes that experience a relatively routine character. Hence, the evaluation of the methodology in small and medium sized enterprises with less routine design processes is pending.

10.2 Implications on Design Practice

The proposed methodology builds on collaborative workshops with designers to identify designers' needs and increase the awareness of designers about design automation opportunities: what are potential design automation use cases and what are the related potentials and implications with respect to design practice. Additionally, by application of standardized graphical modeling languages, the methodology aims at reducing the abstraction of design automation task definition to make the different steps of the methodology accessible to designers and to support validation and communication of yielded results. Hence, different viewpoints from stakeholders involved in design can be taken into account to justify design automation implementation in design practice. In particular, the methodology builds on the idea that designers need to be aware of their design process in order to systematically improve it. The results of the evaluation show that the application of the methodology changes the designers' perception of the design process and design automation opportunities. Further, the application of a graphical modeling language for design automation task formalization was evaluated as beneficial to support validation, communication, maintenance and eventually formalization of design automation tasks.

To systematically support the estimation of the potential impact of design automation on design practice, the methodology includes development of scenarios of the future design processes as well as derivation and quantification of metrics. Focus is put on metrics that enable comprehensive assessment of design automation potential among multiple dimensions of the design process including the tools and technologies. Regarding the estimation of metrics values, the results of evaluation show that a basic understanding of the working principles is mandatory to enable reliable rating of metrics. Altogether, the metrics and models of future scenarios of design processes enable to relate design automation to design practice and assess its potential impact on design performance from the perspective of designers. The proposed methods extend state-of-practice by directly reflecting the needs of designers instead of relying on high-level drivers potentially not reflecting the requirements of design practice. Whereas this procedure potentially leads to some not realized design automation projects, in the long term the reputation of design automation is increased since unfulfilled expectations such as time-savings up to 99% (Reddy, Sridhar, and Rangadu 2015) are mitigated. The qualitative feedback by practitioners indicates that systematic potential estimation is perceived as important, as highlighted by the team leader of Case 3: "If we take this effort [for DA implementation], we also want to measure its impact."

Finally, the methodology emphasizes the close integration of design automation applications to design practice and related PLM. Hence, integration is considered not only from a design process but also tools and technology perspective to reduce duplication of data, rework etc. The analysis of a design process and supporting IT infrastructure as proposed in the method for identification of design automation use cases aims at specification of design automation applications that are tailored to the design process under investigation. This leads to more mature design automation solutions (Willner, Gosling, and Schönsleben 2016) that feature usability based on seamless integration of design automation to design practice. The evaluation shows that application of the methodology supports identification of redundancies in knowledge formalization and specification of interfaces to third party software.

Eventually, the methodology will foster transition of design automation methods from academia to design practice by increased understanding of design automation opportunities and related quantification of the potential impact on design. Further, the enhanced integration in design processes and possibility to formalize design automation tasks will contribute to an increased acceptance of design automation in design practice.

10.3 Implications on Design Research and Contributions

In the following, the implications on design research are highlighted by indicating the achieved contributions in this work.

First contribution: *Consolidation of research related to design automation task definition.*

From the technological point of view, a systematic review and consolidation of literature from the fields KBE and CDS is conducted. The analysis of knowledge levels of addressed design automation task as well as applied knowledge formalization and reasoning techniques enable identification of commonalities and shared research interests for research on design automation methods. In particular, a design automation task categorization that reflects the opportunities offered by state-of-the-art design automation methods is yielded. The categories are characterized by the knowledge levels on input, output and goals and the analysis of design automation methods shows a trend towards application of SysML as a graphical and standardized language for knowledge formalization independent on a specific design automation method.

Second, literature from design process analysis, design performance assessment and knowledge formalization is reviewed and a meta-model for reasoning in the scope of design automation task definition is derived. The meta-model builds upon standardized and graphical languages to feature reuse. In particular, the ArchiMate language is used for capturing the design processes in task-precedence models considering design activities, supporting tools and technologies as well as the motivational aspects of design. Further, the meta-model relates the ArchiMate model to the corresponding design automation task formalization in SysML. Application evaluation shows that the languages are accepted by the designers and support the communication and understanding of information related to design automation task definition.

Second contribution: *A methodology for design automation task definition reflecting the needs of design practitioners and accounting for multiple viewpoints based on collaboration.*

The methodology proposed in this work focuses on collaboration to systematically account for different viewpoints and related needs of designers with different roles in the design process. In fact, the focus on systematic evaluation of the potential impact of design automation on design practice and the early involvement of designers correspond to a major difference of the proposed methodology to available methodologies for design automation implementation, e.g. MOKA (Stokes and MOKA Consortium 2001) or VDI 5610 (Verein Deutscher Ingenieure 2017). Particularly, design automation is linked to design practice by formulation of design automation goals that mitigate potential failure modes in design. Thus, focus is put on design automation task definition from a design practitioners' perspective instead of managers and knowledge engineers.

Further, the methodology is characterized by the idea of design automation task definition independent on a specific design automation method. In particular, the SysML is identified as a means to enable graphical knowledge formalization in a neutral format and established language. Particular focus is put on application of already existing modeling elements within the SysML language in order to rely on established syntax rather than extending the language based on definition of custom stereotypes. Guided by templates accounting for the knowledge levels characterizing a design automation task, design automation task definition independent on the underlying design automation method is enabled. This allows practitioners to view design automation from a knowledge instead of a technology perspective. Results from evaluation of the methodology with industrial cases show that the application of the methodology and supporting methods enables designers to formalize design automation task themselves and changes the awareness of design automation opportunities and related potential.

Third contribution: *the indication of potential for design automation application for the early stages of design.*

The evaluation of the methodology was conducted for three industrial cases. In particular, the application of the method for estimation of the impact of design automation for Case 2 – Hydraulic Systems highlights the potential for design automation to support the early stages of design by generation of alternatives design leading to investigation of different concepts in design practice. These findings contribute to recent surveys where it is shown that the potential for design automation application in the early stages is not acknowledged by designers, e.g. (Rigger and Vosgien 2018).

10.4 Overall Research Question

Based on the characteristics of the provided methodology and its evaluation based on industrial use cases, the overall research question "*What are the key factors to increase design automation application in industry*?" can be answered as follows:

Most importantly, designers need to be able to understand the opportunities of design automation application in design practice. The **viewpoint on design automation** needs to be moved from a technology oriented point of view that focuses on knowledge formalization and reasoning techniques towards a knowledge level point of view. Thus, designers are enabled to identify the design automation use cases and also estimate the related potential it offers.

Next, **designers need to be involved in design automation task definition** to specify design automation applications that address the needs of design practitioners. Designers need to be aware about the details of their design processes and shortcomings to fully understand the value of design automation implementation. The **assessment of potential design automation applications based on metrics** enables to comprehensively assess the implications of design automation implementation. The methodology elaborates on different viewpoints on design based on workshops with designers with different roles, e.g. the team leaders. Therefore, a more comprehensive and realistic view on design practice and potential estimation is attained considering multiple perspectives on design.

Further, design processes also need to be considered from a tools and technology perspective to enable **integration of potential design automation applications to design practice**. The automation of interfaces not only avoids redundancies and inconsistencies in data but also increases usability since rework and manual transformations are mitigated.

Finally, the methodology focuses on a **collaborative approach** for design automation task definition. Collaborative workshops are applied to develop models of design processes, identify use cases, estimate the related potential and formalize design automation tasks. To enable efficient collaboration, the methodology focuses on usage of **standardized languages and graphical modeling** of design processes and knowledge formalization. Thereby, results can be **efficiently communicated** to the different stakeholders involved in industry when it comes to decisions regarding design automation implementation in industrial practice.

10.5 Limitations and Future Work

The application of the methodology and supporting methods show that the proposed systematic works well for small scale design tasks with a relatively routine character so that the micro-level analysis of design processes can be conducted. Hence, future work should focus on the **evaluation for less routine design processes as well as within**

different industries. This readily enables evaluation of the sub-methods as well as the underlying principles of the design automation task categorization, e.g. do the categories reflect design practice and enable identification of opportunities in a broader context of engineering design. Further, the **aspects of collaboration** need to be further investigated: how to select the appropriate team for design automation task definition, how to support interaction etc. The integration of game-playing scenarios for value stream mapping or the design automation task formalization needs to be investigated to increase the motivation for conducting these mentally demanding tasks. Related to the latter, the implementation of software that is tailored to **design automation task formalization** needs to be implemented to increase the related usability. Even though the tool needs to consider the SysML standard, the scope of the modeling environment can be reduced to the expressiveness and capabilities required for design automation task formalization. Further, workflow patterns etc. can be implemented to increase usability and enable design automation task formalization without any support. Also the automated transformation requires further investigation so that the semantic correctness of formalization can be readily assessed. In this respect, future work should address the transformation from the perspective of how to sequence the translation of code since the model does not account for these aspects. Probably, generic schemes for design automation as proposed in (Cagan et al. 2005; Shea and Starling 2003) can be used to organize transformation. However, the task formalization also needs to be conducted for other types of design automation tasks so to evaluate applicability and feasibility of the approach. In particular, the **scalability of the approach** needs to be investigated for large-scale designs. Finally, the **acquisition and externalization of knowledge** from domain experts needs to be investigated. In this work, it is assumed that the graphical representation and indication of interdependencies support domain experts in externalizing their knowledge. Yet, more detailed instigations need to be conducted for evaluation of this working hypothesis.

Regarding the identification of **design automation use cases**, future work should elaborate on the identification of the right **level of granularity to investigate** design processes. In this case, well-defined interfaces of the design process are assumed so that parts of the design process can be analyzed independent on the remaining steps. Yet, potentially knowledge from design processes that are considered out of scope of the workshop need to be taken into account, too. Future research should investigate different strategies for conducting the workshops: the detailed analysis of specific parts of the design process as conducted in this work, or more comprehensive analysis of design processes related to one product.

Related to the analysis of design processes, the integration of design automation from a tools and technology perspective requires further investigation. In this work, focus is put on identification and high-level specification of interfaces. Future work should address the **integration of design automation to PLM strategies** from a more technological and detailed perspective. In this respect, the appropriate level of tool integration needs to be investigated. Both a cost as well as a usability perspective needs to be taken into account.

To further elaborate on potential estimation of design automation, future research needs to address **methods for reliable quantification of metrics**. In this work, focus is put on educated guesses by experts and team discussions to account for different viewpoints. Potentially, metrics quantification based on existing data and prediction models that take into account the specifics of design automation methods can be used to supplement the team discussion. Further, **methods for interpretation and indication of design automation potential** need to be elaborated on, e.g. balanced scorecards (Kaplan and Norton 1990) to enable designers to make informed decisions based on metrics.

From the perspective of considered technologies, the focus was put on design automation methods that focus on reuse of computationally encoded knowledge. Future work should assess the **investigation of other design automation methods**. Efforts from software vendors as well as other fields of research like systems engineering or design by analogy (Chakrabarti et al. 2011) such as case-based reasoning (Kolodner 1993) or machine-learning techniques (McComb 2018) need to be taken into account. Thus, the design automation task categorization can be further enriched.

11 Summary and Conclusion

The work presented in this thesis aims to increase the rate of application of design automation in industry and to foster transition of design automation methods from academia to industry. In order to address these aims, the proposed methodology for design automation task definition systematically builds upon the design automation task categorization that targets at consolidation of the research field design automation from a technology point of view. Design automation tasks as addressed by design automation methods from the fields KBE and CDS are analyzed from a knowledge level perspective. This enables derivation of a categorization that allows viewing design automation from the perspective of required and generated knowledge instead of specific knowledge formalizations and reasoning techniques. Based on design automation task characteristics, the required input product knowledge, goals and output knowledge can be explicitly indicated independent on the underlying design automation method. The evaluation based on three industrial cases shows that the proposed method for identification of design automation opportunities enables design practitioners to grasp the opportunities of design automation based on the knowledge level required for design automation task definition. Hence, design automation opportunities can be identified independent on fundamental knowledge of design automation methods.

In order to further mitigate the gap between the formal representation of design automation tasks and design practice, the methodology focuses on involving designers for design automation task definition. The integration of designers for reengineering design processes and design process analysis shows that the awareness of the design process and related potential for improvement is enhanced. This permits identification of design automation use cases and formulation of design automation goals referring to the removal of potential failure modes in design. Thus, the designers’ needs are identified and linked to design automation.

Building on design automation goals, the methods on derivation of metrics and estimation of the impact of design automation implementation on design practice further enable to systematically evaluate design automation applications in practice. Based on metrics, design automation use cases can be benchmarked and assessed. The evaluation for different design automation tasks in different industrial contexts shows the necessity to conduct a case-specific analysis. The proposed methodology enables derivation of the required metrics and supports rating based on collaborative workshops. The attained results from application show that a comprehensive set of metrics is obtained and designers consider the systematic assessment of design automation potential crucial for informed decisions. The estimation of the impact of design automation for a use case supporting a more conceptual design task highlights the potential for design automation in the early stages of design.

Finally, the proposed methodology features design automation task formalization based on a standardized graphical modeling language. By using design automation task templates and model libraries, the method features modularization and reuse of knowledge, respectively. The expressiveness of the applied modeling language SysML also enables automated transformation of the design automation task formalization to different formalizations according to design automation methods. A user study with participants from industry highlights the potential for graphical formalization to enable formalization and validation of knowledge for design automation task definition by designers.

To sum up, this thesis successfully addresses the aims and objectives posed in the introduction. The proposed systematic not only improves the understanding and perception of design automation in industry but also supports consolidation and alignment of research on design automation. Also, methods are presented that build upon and extend state-of-the-art design automation research with respect to methods for design automation task definition focusing on the needs of design practitioners.

Bibliography

Abramovici, M. 2007. “Future Trends in Product Lifecycle Management (PLM).” In *The Future of Product Development*, edited by Professor Dr-Ing Frank-Lothar Krause, 665–74. Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/978-3-540-69820-3_64.

Amen, Rafael, Ingvar Rask, and Staffan Sunnersjö. 1999. “Matching Design Tasks to Knowledge-Based Software Tools - When Intuition Does Not Suffice.” In *Proceeding of DETC99*, 1165–74. Las Vegas, Nevada.

Anderson, David, K. Blake Perez, Zack Xuereb, Kevin Otto, and Kris Wood. 2018. “Design Processes of Design Automation Practitioners.” In *Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 85436–39. Quebec City: ASME.

Andrae, René, and Peter Köhler. 2016. “Simulation-Oriented Transformation of CAD Models.” *Computer-Aided Design and Applications* 13 (3): 340–47. https://doi.org/10.1080/16864360.2015.1114392.

Andreasen, Mogens Myrup. 2011. “45 Years with Design Methodology.” *Journal of Engineering Design* 22 (5): 293–332. https://doi.org/10.1080/09544828.2010.538040.

Arora, Jasbir. 2004. *Introduction to Optimum Design*. Academic Press.

Asrar-ul-Haq, Muhammad, and Sadia Anwar. 2016. “A Systematic Review of Knowledge Management and Knowledge Sharing: Trends, Issues, and Challenges.” Edited by Tahir Nisar. *Cogent Business & Management* 3 (1). https://doi.org/10.1080/23311975.2015.1127744.

Basili, Victor R. 1992. “Software Modeling and Measurement: The Goal/Question/Metric Paradigm.” *Computer Science Technical Report Series*

Bauch, Christoph. 2004. “Lean Product Development: Making Waste Transparent.” Diploma Thesis, MIT, TUM.

Baxter, David, James Gao, Keith Case, Jenny Harding, Bob Young, Sean Cochrane, and Shilpa Dani. 2007. “An Engineering Design Knowledge Reuse Methodology Using Process Modelling.” *Research in Engineering Design* 18 (1): 37–48. https://doi.org/10.1007/s00163-007-0028-8.

Bayrak, Alparslan Emrah, Yi Ren, and Panos Y. Papalambros. 2016. “Topology Generation for Hybrid Electric Vehicle Architecture Design.” *Journal of Mechanical Design* 138 (8): 081401–10. https://doi.org/10.1115/1.4033656.

Beihoff, B., C. Oster, S. Friedenthal, Paredis, Christiaan, D. Kemp, H. Stoewer, D. Nichols, and J. Wade. 2014. “A World in Motion – Systems Engineering Vision 2025.” San Diego, California: INCOSE.

Bermell-Garcia, P., and I-S. Fan. 2008. “Practitioner Requirements for Integrated Knowledge-Based Engineering in Product Lifecycle Management.” *International Journal of Product Lifecycle Management* 3 (1): 3–20. https://doi.org/10.1504/IJPLM.2008.019968.

Bermell-Garcia, Pablo, Wim J.C. Verhagen, Simon Astwood, Kiran Krishnamurthy, Jean Luc Johnson, Domingo Ruiz, Gary Scott, and Richard Curran. 2012. “A Framework for Management of Knowledge-Based Engineering Applications as Software Services: Enabling Personalization and Codification.” *Advanced Engineering Informatics* 26 (2): 219–30. https://doi.org/10.1016/j.aei.2012.01.006.

Bernus, Peter, Laszlo Nemes, and Günter J. Schmidt. 2003. *Handbook on Enterprise Architecture*. Springer Science & Business Media.

Blessing, Lucienne T. M., and Amaresh Chakrabarti. 2009. *DRM, a Design Research Methodology*. Dordrecht ; London: Springer. https://doi.org/10.1007/978-1-84882-587-1.

Bock, Conrad, Raphael Barbau, Ion Matei, and Mehdi Dadfarnia. 2017. “An Extension of the Systems Modeling Language for Physical Interaction and Signal Flow Simulation: SYSML for physical interaction and signal flow simulation.” *Systems Engineering* 20 (5): 395–431. https://doi.org/10.1002/sys.21380.

Boehnke, Daniel, Axel Reichwein, and Stephan Rudolph. 2009. “Design Language for Airplane Geometries Using the Unified Modeling Language.” In *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 661–670. San Diego, California: American Society of Mechanical Engineers.

Bogusch, Ralf, Sabine Ehrich, Roland Scherer, Tobias Sorg, and Robert Wöhler. 2016. “A Lean Systems Engineering Approach for the Development of Safety-Critical Avionic Systems.” In *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*.

Bolognini, Francesca, Ashwin A. Seshia, and Kristina Shea. 2007. “Exploring the Application of Multidomain Simulation-Based Computational Synthesis Method in MEMS Design.” In *Proceedings of ICED 07*. Paris, France: Design Society.

Bracewell, R. H., K. Shea, P. M. Langdon, L. T. M. Blessing, and P. J. Clarkson. 2001. “A Methodology for Computational Design Tool Research.” In *Proceedings of the 13th International Conference on Engineering Design (ICED’01)*, edited by Kenneth Michael Wallace, Design:181–88. Bury St Edmunds: Professional Engineering Publishing Ltd.

Brandt, Eva, and Jörn Messeter. 2004. “Facilitating Collaboration through Design Games.” In *Proceedings Participatory Design Conference 2004*. Toronto, Canada.

Brinkop, Axel. 2016. “Marktführer Produktkonfiguration.” www.brinkop-consulting.com.

Brown, David C., and Balakrishnan Chandrasekaran. 1990. *Design Problem Solving: Knowledge Structures and Control Strategies*. London; San Mateo, Calif.: Morgan Kaufmann Publishers In.

Bryant, Cari R., Daniel A. McAdams, Robert B. Stone, Tolga Kurtoglu, and Matthew I. Campbell. 2005. “A Computational Technique for Concept Generation.” In *Proceedings of ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2005:267–76. ASME. https://doi.org/10.1115/DETC2005-85323.

Bylund, N, O. Isaksson, V. Kalhori, and T. Larsson. 2004. “Enhanced Engineering Design Practice Using Knowledge Enabled Engineering with Simulation Methods.” In *Proceedings of DESIGN 2004*, 841–46. Dubrovnik, Croatia.

Cagan, Jonathan, Matthew I. Campbell, Susan Finger, and Tetsuo Tomiyama. 2005. “A Framework for Computational Design Synthesis: Model and Applications.” *Journal of Computing and Information Science in Engineering* 5 (3): 171–81. https://doi.org/10.1115/1.2013289.

Cagan, Jonathan, Ignacio E. Grossmann, and John Hooker. 1997. “A Conceptual Framework for Combining Artificial Intelligence and Optimization in Engineering Design.” *Research in Engineering Design* 9 (1): 20–34. https://doi.org/10.1007/BF01607055.

Campbell, Matthew I., Jonathan Cagan, and Kenneth Kotovsky. 1999. “A-Design: An Agent-Based Approach to Conceptual Design in a Dynamic Environment.” *Research in Engineering Design* 11 (3): 172–92. https://doi.org/10.1007/s001630050013.

Canedo, Arquimedes, and Jan H. Richter. 2014. “Architectural Design Space Exploration of Cyber-Physical Systems Using the Functional Modeling Compiler.” *Procedia CIRP* 21: 46–51. https://doi.org/10.1016/j.procir.2014.03.183.

Ćatić, Amer, and Johan Malmqvist. 2007. “Towards Integration of KBE and PLM.” *Proceedings of the International Conference on Engineering Design (ICED07), Paper*, no. 540. http://www.designsociety.org/download-publication/25568/towards_integration_of_kbe_and_plm.

Cederfeldt, Mikael. 2006. “Towards a Strategy for Mapping of Design Problems to Suitable Solutions – A Case of Design Automation Using CBR.” In *Proceedings of DESIGN 06*. International Design Conference, Dubrovnik, Croatia.

Cederfeldt, Mikael, Fredrik Elgh, and others. 2005. “Design Automation in SMEs–Current State, Potential, Need and Requirements.” In *DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design, Melbourne, Australia, 15.-18.08. 2005*.

Chakrabarti, Amaresh, Kristina Shea, Robert Stone, Jonathan Cagan, Matthew Campbell, Noe Vargas Hernandez, and Kristin L. Wood. 2011. “Computer-Based Design Synthesis Research: An Overview.” *Journal of Computing and Information Science in Engineering* 11 (2): 021003–021003. https://doi.org/10.1115/1.3593409.

Chandrasegaran, Senthil K., Karthik Ramani, Ram D. Sriram, Imré Horváth, Alain Bernard, Ramy F. Harik, and Wei Gao. 2013. “The Evolution, Challenges, and Future of Knowledge Representation in Product Design Systems.” *Computer-Aided Design* 45 (2): 204–28. https://doi.org/10.1016/j.cad.2012.08.006.

Chandrasekaran, Balakrishnan. 1990. “Design Problem Solving: A Task Analysis.” *AI Magazine* 11 (4): 59–71.

Chao, Lawrence P., and Kosuke Ishii. 2007. “Design Process Error Proofing: Failure Modes and Effects Analysis of the Design Process.” *Journal of Mechanical Design* 129 (5): 491. https://doi.org/10.1115/1.2712216.

Chapman, Craig B., and Martyn Pinfold. 2001. “The Application of a Knowledge Based Engineering Approach to the Rapid Design and Analysis of an Automotive Structure.” *Advances in Engineering Software* 32 (12): 903–12. https://doi.org/10.1016/S0965-9978(01)00041-2.

Chavali, Siva R., Chiradeep Sen, Gregory M. Mocko, and Joshua D. Summers. 2008. “Using Rule Based Design in Engineer to Order Industry: An SME Case Study.” *Computer-Aided Design and Applications* 5 (1–4): 178–93. https://doi.org/10.3722/cadaps.2008.178-193.

Chen, David, Guy Doumeingts, and François Vernadat. 2008. “Architectures for Enterprise Integration and Interoperability: Past, Present and Future.” *Computers in Industry*, Enterprise Integration and Interoperability in Manufacturing Systems, 59 (7): 647–59. https://doi.org/10.1016/j.compind.2007.12.016.

Cherry, Erin, and Celine Latulipe. 2014. “Quantifying the Creativity Support of Digital Tools through the Creativity Support Index.” *ACM Transactions on Computer-Human Interaction* 21 (4): 1–25. https://doi.org/10.1145/2617588.

Cho, Jullius, Thomas Vosgien, Thorsten Prante, and Detlef Gerhard. 2016. “KBE-PLM Integration Schema for Engineering Knowledge Re-Use and Design Automation.” In *IFIP International Conference on Product Lifecycle Management*, 43–55. Springer. http://link.springer.com/chapter/10.1007/978-3-319-54660-5_5.

Choi, Jin-Woo. 2009. “Architecture of a Knowledge Based Engineering System for Weight and Cost Estimation for a Composite Airplane Structures.” *Expert Systems with Applications* 36 (8): 10828–36. https://doi.org/10.1016/j.eswa.2008.10.049.

Chulvi, Vicente, Alejandro Sancho, David Cebrián, Ricardo Jiménez, Carlos Munoz, and Rosario Vidal. 2007. “Knowledge-Based Engineering in Cranioplasty Implant Design.” In *Proceedings of International Conference on Engineering Design 2007*, 411–12. Paris, France.

CMMI. 2010. “CMMI® for Services.” Technical Report 1.3. Carnegie-Mellon University.

Colombo, Giorgio, Ambrogio Girotti, and Edoardo Rovida. 2005. “Automatic Design of a Press Brake for Sheet Metal Bending.” In , edited by Andrew Samuel and William Lewis, 704–11. Barton ACT: Engineers Australia.

Colombo, Giorgio, Roberto Morotti, Daniele Regazzoni, and Caterina Rizzi. 2015. “An Approach to Integrate Numerical Simulation within KBE Applications.” *International Journal of Product Development* 20 (2): 107–125.

Colombo, Giorgio, Dante Pugliese, Peter Klein, and Johannes Lutzemnberger. 2014. “A Study for Neutral Format to Exchange and Reuse Engineering Knowledge in KBE Applications.” In *Engineering, Technology and Innovation (ICE), 2014 International ICE Conference On*, 1–10. IEEE.

Cooper, Dave, and Gianfranco LaRocca. 2007. “Knowledge-Based Techniques for Developing Engineering Applications in the 21st Century.” In *7th AIAA Aviation Technology, Integration and Operations Conference*, 1–22. Belfast, Northern Ireland: American Institute of Aeronautics and Astronautics.

Coorey, Benjamin P., and Julie R. Jupp. 2014. “Generative Spatial Performance Design System.” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 28 (03): 277–83. https://doi.org/10.1017/S0890060414000225.

Corallo, Angelo, Robert Laubacher, Alessandro Margherita, and Giuseppe Turrisi. 2009. “Enhancing Product Development through Knowledge-Based Engineering (KBE): A Case Study in the Aerospace Industry.” *Journal of Manufacturing Technology Management* 20 (8): 1070–83. https://doi.org/10.1108/17410380910997218.

Costa, Janaina MH, Monica Rossi, Eric Rebentisch, Sergio Terzi, Marco Taisch, and Deborah Nightingale. 2014. “What to Measure for Success in Lean System Engineering Programs?” *Procedia Computer Science* 28: 789–798.

Cross, Nigel, and Anita Clayburn Cross. 1995. “Observations of Teamwork and Social Processes in Design.” *Design Studies* 16 (2): 143–70. https://doi.org/10.1016/0142-694X(94)00007-Z.

Cui, Jin-ju, and De-yu Wang. 2013. “Application of Knowledge-Based Engineering in Ship Structural Design and Optimization.” *Ocean Engineering* 72 (November): 124–39. https://doi.org/10.1016/j.oceaneng.2013.06.013.

Danjou, Stéphane, Norman Lupa, and Peter Koehler. 2008. “Approach for Automated Product Modeling Using Knowledge-Based Design Features.” *Computer-Aided Design & Applications* 5 (5): 622–29.

Di Gironimo, G. 2014. “A Knowledge-Based Engineering Approach for Supporting Railway Manufacturers from the Tender Notice to the Designing Phase.” In *Computers in Railways XIV*, edited by A. Lanzotti, R. Mozzillo, F. Peluso, G. Calvano, S. Puzelli, and P. Verdosci, 125–37. WIT Press. https://doi.org/10.2495/CR140101.

Dinar, Mahmoud, Andreea Danielescu, Christopher MacLellan, Jami J. Shah, and Pat Langley. 2015. “Problem Map: An Ontological Framework for a Computational Study of Problem Formulation in Engineering Design.” *Journal of Computing and Information Science in Engineering* 15 (3): 031007–17. https://doi.org/10.1115/1.4030076.

Dinar, Mahmoud, Jami J. Shah, Jonathan Cagan, Larry Leifer, Julie Linsey, Steven M. Smith, and Noe Vargas Hernandez. 2015. “Empirical Studies of Designer Thinking: Past, Present, and Future.” *Journal of Mechanical Design* 137 (2): 021101. https://doi.org/10.1115/1.4029025.

Dinar, Mahmoud, Joshua D. Summers, Jami Shah, and Yong-Seok Park. 2016. “Evaluation of Empirical Design Studies and Metrics.” In *Experimental Design Research*, edited by Philip Cash, Tino Stanković, and Mario Štorga, 13–39. Cham: Springer International Publishing. http://link.springer.com/10.1007/978-3-319-33781-4_2.

Do, Namchul. 2014. “Application of OLAP to a PDM Database for Interactive Performance Evaluation of In-Progress Product Development.” *Computers in Industry* 65 (4): 636–45. https://doi.org/10.1016/j.compind.2014.01.014.

Douglass, Bruce Powel. 2016. *Agile Systems Engineering*. 1st ed. Elsevier. https://doi.org/10.1016/C2014-0-02102-8.

Duffy, Alex H. B. 2005. “Design Process and Performance.” In *Engineering Design-Theory and Practice*, by P. J. Clarkson and M. J. Huhtala, 76–85. A Symposium in Honour of Ken Wallace. Cambridge, U.K.

Dym, Clive L, and David C Brown. 2012. *Engineering Design: Representation and Reasoning*. 2nd ed. New York: Cambridge University Press.

Dym, Clive L., and Raymond E. Levitt. 1991. *Knowledge-Based Systems in Engineering*. New York: McGraw-Hill.

Eckert, C. M., and M. K. Stacey. 2010. “What Is a Process Model? Reflections on the Epistemology of Design Process Models.” In *Modelling and Management of Engineering Processes*, edited by Peter Heisig, P. John Clarkson, and Sandor Vajna, 3–14. Springer London. http://link.springer.com/chapter/10.1007/978-1-84996-199-8_1.

Elgh, Fredrik. 2007. “Modelling and Management of Manufacturing Requirements in Design Automation Systems.” In *Complex Systems Concurrent Engineering*, edited by Geilson Loureiro BSc MSc and Richard Curran BEng, 321–28. Springer London. http://link.springer.com/chapter/10.1007/978-1-84628-976-7_36.

Elst, S. W. G. van der, and M. J. L. van Tooren. 2008. “Application of a Knowledge Engineering Process to Support Engineering Design Application Development.” In *Collaborative Product and Service Life Cycle Management for a Sustainable World*, edited by Richard Curran, Shuo-Yan Chou, and Amy Trappey, 417–31. Advanced Concurrent Engineering. Springer London. http://link.springer.com/chapter/10.1007/978-1-84800-972-1_39.

Emberey, C.L., N.R Milton, J.P.T.J. Berends, M.J.L van Tooren, and S.W.G van der Elst. 2007. “Application of Knowledge Engineering Methodologies to Support Engineering Design Application Development in Aerospace.” In *7th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*. Belfast, Northern Ireland. https://doi.org/10.2514/6.2007-7708.

Eppinger, Steven D, and Tyson R. Browning. 2012. *Design Structure Matrix Methods and Applications*. Cambridge, Mass.: MIT Press.

Fan, Ip-Shing, and Pablo Bermell-Garcia. 2008. “International Standard Development for Knowledge Based Engineering Services for Product Lifecycle Management.” *Concurrent Engineering* 16 (4): 271–77. https://doi.org/10.1177/1063293X08100027.

Faulkner, Laura. 2003. “Beyond the Five-User Assumption: Benefits of Increased Sample Sizes in Usability Testing.” *Behavior Research Methods, Instruments, & Computers* 35 (3): 379–83. https://doi.org/10.3758/BF03195514.

Felfernig, Alexander, Gerhard E. Friedrich, and Dietmar Jannach. 2000. “UML as Domain Specific Language for the Construction of Knowledge-Based Configuration Systems.” *International Journal of Software Engineering and Knowledge Engineering* 10 (04): 449–69. https://doi.org/10.1142/S0218194000000249.

Fenton, Norman, and James Bieman. 2014. *Software Metrics: A Rigorous and Practical Approach*. Third Edition. Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series. CRC Press. https://doi.org/10.1201/b17461.

Frank, G., and C. Hillbrand. 2012. “Automatic Support of Standardization Processes in Design Models.” In *16th International Conference on Intelligent Engineering Systems (INES)*, 393–98. https://doi.org/10.1109/INES.2012.6249865.

Frank, Gerald, Doris Entner, Thorsten Prante, Vaheh Khachatouri, and Martin Schwarz. 2014. “Towards a Generic Framework of Engineering Design Automation for Creating Complex CAD Models.” *International Journal On Advances in Systems and Measurements* 7 (1 and 2): 179–92.

Friedenthal, Sanford, Regina Griego, and Mark Sampson. 2007. “INCOSE Model Based Systems Engineering (MBSE) Initiative.” In *INCOSE 2007 Symposium*.

Friedenthal, Sanford, Alan Moore, and Rick Steiner. 2014. *A Practical Guide to SysML: The Systems Modeling Language*. Morgan Kaufmann..

Garcia, Patrick, and John Drogosz. 2007. “Lean Engineering-Best Practice in the Automotive Industry.” SAE Technical Paper.

Gautam, Naveen, and Nanua Singh. 2008. “Lean Product Development: Maximizing the Customer Perceived Value through Design Change (Redesign).” *International Journal of Production Economics* 114 (1): 313–332.

Gauthier, Jean-Marie, Fabrice Bouquet, Ahmed Hammad, and Fabien Peureux. 2015. “Tooled Process for Early Validation of SysML Models Using Modelica Simulation.” In *Fundamentals of Software Engineering*, edited by Mehdi Dastani and Marjan Sirjani, 9392:230–37. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-24644-4_16.

Gerhard, Detlef, and C. Lutz. 2011. “IT-Based Configuration and Dimensioning of Customer Specific Products – Towards a Framework for Implementing Knowledge Based Design Assistant Systems.” In *Proceedings of ICED 11*, 192–99. Kobenhavn (DK): Design Society.

Gero, John S. 2000. “Computational Models of Innovative and Creative Design Processes.” *Technological Forecasting and Social Change* 64 (2–3): 183–96. https://doi.org/10.1016/S0040-1625(99)00105-5.

Gero, John S., and Udo Kannengiesser. 2004. “The Situated Function–Behaviour–Structure Framework.” *Design Studies* 25 (4): 373–91. https://doi.org/10.1016/j.destud.2003.10.010.

Gmeiner, Thomas, and Kristina Shea. 2013. “An Ontology for the Autonomous Reconfiguration of a Flexible Fixture Device.” *Journal of Computing and Information Science in Engineering* 13 (2): 021003. https://doi.org/10.1115/1.4023587.

Gries, Bruno, and John Restrepo. 2011. “KPI Measurement in Engineering Design–A Case Study.” In *DS 68-1: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 1: Design Processes, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011*.

Hart, Sandra G., and Lowell E. Staveland. 1988. “Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research.” In *Advances in Psychology*, 52:139–83. Elsevier. https://doi.org/10.1016/S0166-4115(08)62386-9.

Heisserman, Jeff, Raju Mattikalli, and Sean Callahan. 2004. “A Grammatical Approach to Design Generation and Its Application to Aircraft Systems.” In *Proceedings of Generative CAD Systems Symposium*, 4:12–14. Pittsburgh, Pennsylvania.

Heitlager, Ilja, Tobias Kuipers, and Joost Visser. 2007. “A Practical Model for Measuring Maintainability.” In *Quality of Information and Communications Technology, 2007. QUATIC 2007. 6th International Conference on The*, 30–39. IEEE.

Helms, Bergen, and Kristina Shea. 2012. “Computational Synthesis of Product Architectures Based on Object-Oriented Graph Grammars.” *ASME Journal of Mechanical Design* 134 (2): 021008.

Hicks, B. J., S. J. Culley, R. D. Allen, and G. Mullineux. 2002. “A Framework for the Requirements of Capturing, Storing and Reusing Information and Knowledge in Engineering Design.” *International Journal of Information Management* 22 (4): 263–80. https://doi.org/10.1016/S0268-4012(02)00012-9.

Hoisl, Frank, and Kristina Shea. 2013. “Three-Dimensional Labels: A Unified Approach to Labels for a General Spatial Grammar Interpreter.” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1–18.

Hooshmand, Amir, and Matthew I. Campbell. 2014. “Layout Synthesis of Fluid Channels Using Generative Graph Grammars.” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 28 (03): 239–57. https://doi.org/10.1017/S0890060414000201.

Horváth, Péter, Ronald Gleich, and Mischa Seiter. 2015. *Controlling*. Vahlen. https://doi.org/10.15358/9783800649556.

Hotz, Lothar, Alexander Felfernig, Markus Stumptner, Anna Ryabokon, Claire Bagley, and Katharina Wolter. 2014. “Configuration Knowledge Representation and Reasoning.” In *Knowledge-Based Configuration*, 41–72. Elsevier.

Hubka, Vladimir, and W. Ernst Eder. 1982. *Principles of Engineering Design*. 1st English ed. London ; Boston: Butterworth Scientific.

Hutcheson, Ryan S., Robert L. Jordan, and Robert B. Stone. 2006. “Application of a Genetic Algorithm to Concept Variant Selection.” In *Proceedings of IDETC/CIE 2006*, 43–52. Philadelphia, Pennsylvania: Design Society.

Hvam, Lars, Haug Anders, and Henrik Mortensen Niels. 2010. “Assessment of Benefits from Product Configuration Systems.” In *13th Workshop on Configuration, ECAI 2010*.

ISO/IEC. 2011. “ISO/IEC 9126-1: Software Engineering - Product Quality - Part 1: Quality Model.” https://www.iso.org/standard/22749.html.

———. 2015. “ISO/IEC 25024: Systems and Software Quality Requirements and Evaluation (SQuaRE) — Measurement of Data Quality.”

———. 2016a. “ISO/IEC 25022: Systems and Software Quality Requirements and Evaluation (SQuaRE) — Measurement of Quality in Use.”

———. 2016b. “ISO/IEC 25023: Systems and Software Quality Requirements and Evaluation (SQuaRE) — Measurement of System and Software Product Quality.”

ISO/IEC/IEEE. 2011. “ISO/IEC/IEEE 42010 - Systems and Software Engineering -- Architecture Description.”

Jensen, Thomas. 1999. “Functional modelling in a design support system: contribution to a designer’s workbench.” Lyngby: Department of Control and Engineering Design, Technical University of Denmark.

Jin, Yan, and Wei Li. 2007. “Design Concept Generation: A Hierarchical Coevolutionary Approach.” *Journal of Mechanical Design* 129 (10): 1012–22. https://doi.org/10.1115/1.2757190.

Johansson, Joel. 2012. “Combining Case Based Reasoning and Shape Matching Based on Clearance Analyzes to Support the Reuse of Components.” In , 3:603–10. Chicago, Illinois, USA: ASME. https://doi.org/10.1115/DETC2012-70631.

Johansson, Joel, and Fredrik Elgh. 2013. “How to Successfully Implement Automated Engineering Design Systems: Reviewing Four Case Studies.” In , 173–182. Melbourne. https://doi.org/10.3233/978-1-61499-302-5-173.

Kaplan, Robert S., and David Norton. 1990. “The Balanced Scorecard: Measures That Drive Performance.” *Harvard Business Review* 70 (1): 71–79.

Kerzhner, Aleksandr. 2012. “Using Logic-Based Approaches to Explore System Architectures for Systems Engineering.” Doctoral thesis, Georgia: Georgia Institute of Technology.

Kitamura, Yoshinobu, and Riichiro Mizoguchi. 2004. “Ontology-Based Systematization of Functional Knowledge.” *Journal of Engineering Design* 15 (4): 327–51. https://doi.org/10.1080/0954482041000169716З.

Klein, Patrick, Johannes Lützenberger, and Klaus-Dieter Thoben. 2015. “A Proposal for Knowledge Formailization in Product Development Processes.” In *Proceedings of the INternational Conference on Engineering Design, ICED15*, 261–72. Milano, Italy: Design Society.

Kolodner, Janet L. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

Komoto, Hitoshi, and Tetsuo Tomiyama. 2012. “A Framework for Computer-Aided Conceptual Design and Its Application to System Architecting of Mechatronics Products.” *Computer-Aided Design* 44 (10): 931–46. https://doi.org/10.1016/j.cad.2012.02.004.

Königseder, Corinna. 2015. “A Methodology for Supporting Design Grammar Development and Application in Computational Design Synthesis.” Dissertation, Zürich: ETH Zürich.

Königseder, Corinna, Kristina Shea, and Matthew I. Campbell. 2013. “Comparing a Graph-Grammar Approach to Genetic Algorithms for Computational Synthesis of PV Arrays.” In *CIRP Design 2012*, edited by Amaresh Chakrabarti, 105–14. London: Springer London. http://link.springer.com/10.1007/978-1-4471-4507-3_11.

Koziolek, Heiko. 2008. “Goal, Question, Metric.” In *Dependability Metrics*, 39–42. Springer.

Kreimeyer, Matthias, and Udo Lindemann. 2011. *Complexity Metrics in Engineering Design: Managing the Structure of Design Processes*. Springer Science & Business Media.

Kruse, Benjamin. 2016. “A Library-Based Concept Design Approach for Multi-Disciplinary Systems in SysML.” Dissertation, Zürich: ETH Zürich.

Kulon, J., P. Broomhead, and D. J. Mynors. 2006. “Applying Knowledge-Based Engineering to Traditional Manufacturing Design.” *The International Journal of Advanced Manufacturing Technology* 30 (9–10): 945–51. https://doi.org/10.1007/s00170-005-0067-0.

Kurtoglu, Tolga, and M. I. Campbell. 2009. “Automated Synthesis of Electromechanical Design Configurations from Empirical Analysis of Function to Form Mapping.” *Journal of Engineering Design* 20 (1): 83–104. https://doi.org/10.1080/09544820701546165.

La Rocca, Gianfranco. 2012. “Knowledge Based Engineering: Between AI and CAD. Review of a Language Based Technology to Support Engineering Design.” *Advanced Engineering Informatics* 26 (2): 159–79. https://doi.org/10.1016/j.aei.2012.02.002.

La Rocca, Gianfranco, and M. J. L. van Tooren. 2010. “Knowledge-Based Engineering to Support Aircraft Multidisciplinary Design and Optimization.” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 224 (9): 1041–55. https://doi.org/10.1243/09544100JAERO592.

Lankhorst, Marc. 2017. *Enterprise Architecture at Work*. 4th ed. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-01310-2.

Likert, R. 1932. “A Technique for the Measurement of Attitudes.” *Archives of Psychology* 22 140: 55–55.

Lin, Yi-shih, Kristina Shea, Aylmer Johnson, John Coultate, and Jamie Pears. 2009. “A Method and Software Tool for Automated Gearbox Synthesis.” In *Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 5:111–21. San Diego, California.

Lund, Jonathan G., Nathaniel L. Fife, and C. Gregory Jensen. 2005. “PLM-Based Parametrics for Design Automation and Optimization.” *Computer-Aided Design and Applications* 2 (1–4): 37–45. https://doi.org/10.1080/16864360.2005.10738351.

Mandorli, Ferrucio, Sergio Berti, and Michele Germani. 2002. “A KBE System to Manage the Module Configuration Using the Corporate Knowledge.” In *Proceedings of DESIGN 2002*, 385–90. Dubrovnik, Croatia.

Mayer, Richard J., C.P. Menzel, M.K. Painter, P.S. Dewitte, T. Blinn, and B. Perakath. 1995. “Information Integration for Concurrent Engineering (IICE).” DEF3 process description capture method report. KBSI-IICE-90-STR-01-0592-02. KNOWLEDGE BASED SYSTEMS INC COLLEGE STATION TX.

McComb, Christopher. 2018. “Towards the Rapid Design of Engineered Systems Through Deep Neural Networks.” In *Proceedings of DCC 2018*, 3–22. Milano, Italy: Springer.

McKendry, Danny, R.I. Whitfield, and A.H.B. Duffy. 2015. “Business and Information Management Architectures for Delivering Product Lifecycle Management (PLM) in Engineering To Order (ETO) Products.” In *International Conference in Compuer Applications for Shipbuilding*. Bremen: RINA Conference Proceedings.

McManus, H. 2005. *Product Development Value Stream Mapping–(PDVSM) Manual*. The Lean Aerospace Initiative. Cambridge, Mass.

Melao, Nuno, and Michael Pidd. 2000. “A Conceptual Framework for Understanding Business Processes and Business Process Modelling.” *Information Systems Journal* 10 (2): 105–29. https://doi.org/10.1046/j.1365-2575.2000.00075.x.

Ming, Zhenjun, Anand Bali Nellipalli, Yan Yan, Wang, Guoxin, Goh, Chung Hyun, Allen, Janet K., and Mistree, Farrokh. 2018. "PDSIDES—A Knowledge-Based Platform for Decision Support in the Design of Engineering Systems," no. 18: 041001–1. https://doi.org/10.1115/1.4040461.

Ming, Zhenjun, Yan Yan, Guoxin Wang, Jitesh Panchal, Chung Hyun Goh, Janet K. Allen, and Farrokh Mistree. 2015. "Ontology-Based Executable Design Decision Template Representation and Reuse." In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, V01BT02A031–V01BT02A031. Boston, MA: American Society of Mechanical Engineers.

Morgan, James M., and Jeffrey K. Liker. 2006. *The Toyota Product Development System*. Vol. 13533. New York: Productivity Press.

Moullec, Marie-Lise, Marc Bouissou, Marija Jankovic, Jean-Claude Bocquet, François Réquillard, Olivier Maas, and Olivier Forgeot. 2013. "Toward System Architecture Generation and Performances Assessment Under Uncertainty Using Bayesian Networks." *Journal of Mechanical Design* 135 (4): 041002.

Mulder, Bram, Gianfranco La Rocca, J. Schut, and Verhagen, Wim J. C. 2015. "A Methodological Approach for the Optimisation of the Product Development Process by the Application of Design Automation." In *CEAS 2015*. Delft: CEAS.

Münzer, Clemens, Bergen Helms, and Kristina Shea. 2013. "Automatically Transforming Object-Oriented Graph-Based Representations Into Boolean Satisfiability Problems for Computational Design Synthesis." *Journal of Mechanical Design* 135 (10): 101001.

Münzer, Clemens, and Kristina Shea. 2015. "A Simulation-Based CDS Approach: Automated Generation of Simulation Models Based From Generated Concept Model Graphs." In *Proceedings of the ASME 2015 International Design Engineering Technical Conference & Computers and Information in Engineering Conference*, 47353. Boston, MA: ASME. https://doi.org/10.1115/1.4036567.

———. 2017. "Simulation-Based Computational Design Synthesis Using Automated Generation of Simulation Models From Concept Model Graphs." *Journal of Mechanical Design* 139 (7): 071101. https://doi.org/10.1115/1.4036567.

Neubert, Susanne. 1993. "Model Construction in MIKE (Model Based and Incremental Knowledge Engineering)." In *Knowledge Acquisition for Knowledge-Based Systems*, edited by N. Aussenac, G. Boy, B. Gaines, M. Linster, J. -G. Ganascia, and Y. Kodratoff, 723:200–219. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-57253-8_55.

Newell, Allen. 1981. “The Knowledge Level.” *AI Magazine* 2 (2): 1–21. https://doi.org/10.1609/aimag.v2i2.99.

Neyrinck, Adrian, Armin Lechler, and Alexander Verl. 2015. “Automatic Variant Configuration and Generation of Simulation Models for Comparison of Plant and Machinery Variants.” *Procedia CIRP*, The 22nd CIRP Conference on Life Cycle Engineering, 29: 62–67. https://doi.org/10.1016/j.procir.2015.02.069.

Nielsen, Jakob. 1993. *Usability Engineering*. Boston: Academic Press.

———. 1994. “Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier.” In *Cost-Justifiying Usability*, edited by Randolph G. Bias and Clare-Marie Karat, 245–72.

O’Donnell, F.J., and A.H.B. Duffy. 2002. “Modelling Design Development Performance.” *International Journal of Operations & Production Management* 22 (11): 1198–1221. https://doi.org/10.1108/01443570210450301.

Pahl, Gerhard, W. Beitz, J. Feldhusen, and K. H. Grote. 2007. *Engineering Design: A Systematic Approach*. Translated by Ken Wallace and Lucienne T. M. Blessing. Springer. 10.1007/978-1-84628-319-2.

Pal, Pralay, and Kunal K. Ghosh. 2017. “Estimating Digitization Efforts of Complex Product Realization Processes.” *The International Journal of Advanced Manufacturing Technology*. https://doi.org/10.1007/s00170-017-1442-3.

Panchal, J., S. Ferguson, B. DuPont, and J. Allison. 2015. “New Perspectives on Design Automation: Celebrating the 40th Anniversary of the ASME Design Automation Conference.” *Journal of Mechanical Design* 137 (5): 050301–050301. https://doi.org/10.1115/1.4030256.

Panchal, Jitesh H., Marco Gero Fernández, Christiaan J. J. Paredis, Janet K. Allen, and Farrokh Mistree. 2009. “A Modular Decision-Centric Approach for Reusable Design Processes.” *Concurrent Engineering: Research and Applications* 17 (1).

Parks, Moon, Sangeeta Bansal, and David Zilberman. 2016. “Fit-Risk in Development Projects: Role of Demonstration in Technology Adoption.” *Environment and Development Economics* 21 (06): 742–66. https://doi.org/10.1017/s1355770x16000188.

Patel, Jay, and Matthew I. Campbell. 2010. “An Approach to Automate and Optimize Concept Generation of Sheet Metal Parts by Topological and Parametric Decoupling.” *ASME Journal of Mechanical Design* 132: 051001.

Peak, Russell S., Roger M. Burkhart, Sanford A. Friedenthal, Miyako W. Wilson, Manas Bajaj, and Injoong Kim. 2007a. “Simulation-Based Design Using SysML Part 1: A

Parametrics Primer.” In *INCOSE International Symposium*, 17:1516–1535. San Diego, California: Wiley Online Library. http://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2007.tb02964.x/abstract.

———. 2007b. “Simulation-Based Design Using SysML Part 2: Celebrating Diversity by Example.” In *INCOSE International Symposium*, 17:1536–1557. San Diego, California: Wiley Online Library. http://onlinelibrary.wiley.com/doi/10.1002/j.2334-5837.2007.tb02965.x/abstract.

Plietz, Matthias. 2010. “Structured Development Process of Configuration Models.” In *Workshop on Configuration*. Lisbon, Portugal: Lothar Hotz, Alois Haselböck.

Raffaeli, Roberto, Maura Mengoni, and Michele Germani. 2013. “Improving the Link between Computer-Assisted Design and Configuration Tools for the Design of Mechanical Products.” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27 (01): 51–64. https://doi.org/10.1017/S0890060412000388.

Reddy, E. Jayakiran, C. N. V. Sridhar, and V. Pandu Rangadu. 2015. “Knowledge Based Engineering: Notion, Approaches and Future Trends.” *American Journal of Intelligent Systems* 5 (1): 1–17.

Rigger, Eugen, Clemens Münzer, and Kristina Shea. 2016. “Estimating the Potential of State of the Art Design Automation - Tasks, Methods, and Benefits.” In *Proceedings of DESIGN 2016*, 421–32. Dubrovnik, Croatia. https://doi.org/10.1115/1.4036567.

Rigger, Eugen, and Thomas Vosgien. 2018. “Design Automation State of Practice - Potential and Opportunities.” In *Proceedings of the 15th International DESIGN Conference*. https://doi.org/10.21278/idc.2018.0537.

Robinson, Mark A. 2009. “Work Sampling: Methodological Advances and New Applications.” *Human Factors and Ergonomics in Manufacturing*, n/a-n/a. https://doi.org/10.1002/hfm.20186.

———. 2016. “Quantitative Research Principles and Methods for Human-Focused Research in Engineering Design.” In *Experimental Design Research*, 1st ed. Springer International Publishing Switzerland. https://doi.org/10.1007/978-3-319-33781-4_3.

Ruschitzka, Margot, Adam Suchodolski, and Jerzy Wróbel. 2010. “Ontology-Based Approach in Hybrid Engineering Knowledge Representation for Stamping Die Design.” In *New World Situation: New Directions in Concurrent Engineering*, 227–35. Advanced Concurrent Engineering. Springer London.

Schmidt, Linda C., and Jonathan Cagan. 1997. “GGREADA: A Graph Grammar-Based Machine Design Algorithm.” *Research in Engineering Design* 9 (4): 195–213. https://doi.org/10.1007/BF01589682.

Schmidt, Linda C., Harshawardhan Shetty, and Scott C. Chase. 2000. “A Graph Grammar Approach for Structure Synthesis of Mechanisms.” *Journal of Mechanical Design* 122 (4): 371. https://doi.org/10.1115/1.1315299.

Schotborgh, Wouter O., Chris McMahon, and Fred J.A.M. Van Houten. 2012. “A Knowledge Acquisition Method to Model Parametric Engineering Design Processes.” *International Journal of Computer Aided Engineering and Technology* 4 (4): 373–91. https://doi.org/10.1504/IJCAET.2012.047812.

Schreiber, Guus, Bob Wielinga, Hans Akkermans, Walter Van de Velde, and Anjo Anjewierden. 1994. “CML: The CommonKADS Conceptual Modelling Language.” In *International Conference on Knowledge Engineering and Knowledge Management*, 1–25. Springer. http://link.springer.com/chapter/10.1007/3-540-58487-0_1.

Schut, E. J., S. Kosman, and R. Curran. 2013. “A Value Scan Methodology to Improve Industrial Operations.” In *Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment*, edited by Josip Stjepandić, Georg Rock, and Cees Bil, 411–23. London: Springer London. http://link.springer.com/10.1007/978-1-4471-4426-7_36.

Seffah, Ahmed, Mohammad Donyaee, Rex B. Kline, and Harkirat K. Padda. 2006. “Usability Measurement and Metrics: A Consolidated Model.” *Software Quality Journal* 14 (2): 159–178.

Shah, Aditya A., Christiaan J. J. Paredis, Roger Burkhart, and Dirk Schaefer. 2012. “Combining Mathematical Programming and SysML for Automated Component Sizing of Hydraulic Systems.” *Journal of Computing and Information Science in Engineering* 12 (4): 041006. https://doi.org/10.1115/1.4007764.

Shea, Kristina, Robert Aish, and Marina Gourtovaia. 2005. “Towards Integrated Performance-Driven Generative Design Tools.” *Automation in Construction* 14 (2): 253–64. https://doi.org/10.1016/j.autcon.2004.07.002.

Shea, Kristina, John Cagan, and Steven J. Fenves. 1997. “A Shape Annealing Approach to Optimal Truss Design With Dynamic Grouping of Members.” *Journal of Mechanical Design* 119 (3): 388. https://doi.org/10.1115/1.2826360.

Shea, Kristina, and Ian F. C. Smith. 2006. “Improving Full-Scale Transmission Tower Design through Topology and Shape Optimization.” *Journal of Structural Engineering* 132 (5): 781–90. https://doi.org/10.1061/(ASCE)0733-9445(2006)132:5(781).

Shea, Kristina, and Alex Starling. 2003. “From Discrete Structures to Mechanical Systems: A Framework for Creating Performance-Based Parametric Synthesis Tools.” In *Technical Report*. AAAI.

Siddique, Zahed, and David W. Rosen. 1999. “Product Platform Design: A Graph Grammar Approach.” In *Proceedings of DETC’99*. Las Vegas, Nevada.

Sim, Siang Kok, and Alex H. B. Duffy. 2003. “Towards an Ontology of Generic Engineering Design Activities.” *Research in Engineering Design* 14 (4): 200–223. https://doi.org/10.1007/s00163-003-0037-1.

Škec, Stanko, Philip Cash, and Mario Štorga. 2017. “A Dynamic Approach to Real-Time Performance Measurement in Design Projects.” *Journal of Engineering Design* 28 (4): 255–86. https://doi.org/10.1080/09544828.2017.1303665.

Stanković, Tino, Mario Štorga, Kristina Shea, and Dorian Marjanović. 2012. “Formal Modelling of Technical Processes and Technical Process Synthesis.” *Journal of Engineering Design* 24 (3): 211–38. https://doi.org/10.1080/09544828.2012.722193.

Starling, Alex C., and Kristina Shea. 2005. “A Parallel Grammar for Simulation-Driven Mechanical Design Synthesis.” In , 85414. Long Beach CA: ASME conference proceedings. http://link.aip.org/link/abstract/ASMECP/v2005/i4739X/p427/s1.

Stjepandić, Josip, Wim J. C. Verhagen, Harald Liese, and Pablo Bermell-Garcia. 2015. “Knowledge-Based Engineering.” In *Concurrent Engineering in the 21st Century*, edited by Josip Stjepandić, Nel Wognum, and Wim J. C. Verhagen, 255–86. Zürich: Springer International Publishing. http://link.springer.com/chapter/10.1007/978-3-319-13776-6_10.

Stokes, Melody, and MOKA Consortium. 2001. *Managing Engineering Knowledge: MOKA: Methodology for Knowledge Based Engineering Applications*. Professional Engineering Publ.

Stone, Robert B., and Kristin L. Wood. 2000. “Development of a Functional Basis for Design.” *Journal of Mechanical Design* 122 (4): 359. https://doi.org/10.1115/1.1289637.

Summers, Joshua D., and Jami J. Shah. 2010. “Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability.” *Journal of Mechanical Design* 132 (2): 021004–021004. https://doi.org/10.1115/1.4000759.

Sunnersjö, Staffan, Mikael Cederfeldt, Fredrik Elgh, and Ingvar Rask. 2006. “A Transparent Design System for Iterative Product Development.” *Journal of Computing and Information Science in Engineering* 6 (3): 300. https://doi.org/10.1115/1.2218363.

Tarkian, Mehdi, Johan Ölvander, Xialong Feng, and Marcus Petterson. 2011. “Product Platform Automation for Optimal Configuration of Industrial Robot Families.” In *Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design*, 4:1–10. Copenhagen, Denmark.

The Open Group. 2009. *The Open Group Architecture Framework (TOGAF)*. Vol. 1. Reading, United Kingdom: The Open Group.

———. 2016. *ArchiMate 3.0 Specification*. Reading, United Kingdom: The Open Group. http://pubs.opengroup.org/architecture/archimate3-doc/.

Thomke, Stefan, and Takahiro Fujimoto. 2000. "The Effect of 'Front-Loading' Problem-Solving on Product Development Performance." *Journal of Product Innovation Management* 17 (2): 128–42.

Tong, Christopher, and D. Sriram. 1992. *Artificial Intelligence in Engineering Design*. Vol. Volume III: Knowledge Acquisition, Commercial Systems, and Integrated Environments. San Diego, California: Academic Press.

Tribelsky, Effi, and Rafael Sacks. 2011. "An Empirical Study of Information Flows in Multidisciplinary Civil Engineering Design Teams Using Lean Measures." *Architectural Engineering and Design Management* 7 (2): 85–101.

Tsai, Yi-Lung, Chun-Fong You, Jhen-Yang Lin, and Kun-Yu Liu. 2010. "Knowledge-Based Engineering for Process Planning and Die Design for Automotive Panels." *Computer-Aided Design and Applications* 7 (1): 75–87. https://doi.org/10.3722/cadaps.2010.75-87.

Tyagi, Satish, Alok Choudhary, Xianming Cai, and Kai Yang. 2015. "Value Stream Mapping to Reduce the Lead-Time of a Product Development Process." *International Journal of Production Economics* 160: 202–212.

Ulrich, Karl T, and Steven D Eppinger. 2004. *Product Design and Development*. Boston: McGraw-Hill/Irwin.

Van der Laan, A.H., and M.J.L. van Tooren. 2005. "Parametric Modeling of Movables for Structural Analysis." *Journal of Aircraft* 42 (6): 1605–13.

Van der Laan, Ton, Brent Vermeulen, and Frank van Dalen. 2008. "Automating the Development Process of Thermoplastic Composite Sheet Ribs Using Knowledge Based Engineering Techniques." In *ICAS 2008*. Anchorage.

van Dijk, Reinier E. C., Xiaojia Wang, Haiqiang, and van Dalen, Frank. 2012. "Multidisciplinary Design and Optimization Framework for Aircraft Box Structures." In *Proceedings of the 3rd Aircraft Structural Design Conference*. Delft: Royal Aeronautical Society.

Velden, Christian van der, Cees Bil, and Xinghuo Xu. 2012. "Adaptable Methodology for Automation Application Development." *Advanced Engineering Informatics* 26 (2): 231–50. https://doi.org/10.1007/978-1-4471-4426-7_36.

Verein Deutscher Ingenieure. 2009. “VDI 5610 - Blatt 1 - Wissensmanagement Im Ingenieurswesen - Grundlagen, Konzepte, Vorgehen.” VDI Guideline.

———. 2017. “VDI 5610 - Blatt 2 - Wissensmanagement Im Engeineering - Wissensbasierte Konstruktion (KBE).” VDI Guideline.

Verhagen, Wim J. C., Bastiaan de Vrught, Joost Schut, and Richard Curran. 2015. “A Method for Identification of Automation Potential through Modelling of Engineering Processes and Quantification of Information Waste.” *Advanced Engineering Informatics* 29: 307–21. https://doi.org/10.1016/j.aei.2015.03.003.

Verhagen, Wim J.C., Pablo Bermell-Garcia, Reinier E.C. van Dijk, and Richard Curran. 2012. “A Critical Review of Knowledge-Based Engineering: An Identification of Research Challenges.” *Advanced Engineering Informatics* 26 (1): 5–15. https://doi.org/10.1016/j.aei.2011.06.004.

Vosgien, Thomas. 2015. “Model-Based System Engineering Enabling Design-Analysis Data Integration in Digital Design Environments : Application to Collaborative Aeronautics Simulation-Based Design Process and Turbojet Integration Studies.” Châtenay-Malabry, Ecole centrale de Paris.

Vosgien, Thomas, Marija Jankovic, Benoit Eynard, Thomas Nguyen Van, and Jean-Claude Bocquet. 2011. “Lean Approach to Integrate Collaborative Product Development Processes and Digital Engineering Systems.” In *DS 68-1: Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 1: Design Processes, Lyngby/Copenhagen, Denmark, 15.-19.08*.

Vosgien, Thomas, Thomas Nguyen Van, Marija Jankovic, Benoît Eynard, and Jean-Claude Bocquet. 2012. “Towards Model-Based System Engineering for Simulation-Based Design in Product Data Management Systems.” In *Product Lifecycle Management. Towards Knowledge-Rich Enterprises*, 612–22. Springer.

Wang, L., X. G. Ming, F. B. Kong, D. Li, and P. P. Wang. 2011. “Focus on Implementation: A Framework for Lean Product Development.” *Journal of Manufacturing Technology Management* 23 (1): 4–24.

Weilguny, L.J., and D. Gerhard. 2009. “Implementation of a Knowledge-Based Engineering Concept Using Feature Technology at an Automotive OEM.” In *Proceedings of ICED 09*, 1–12. Palo Alto.

White, Stephen A., and Derek Miers. 2008. *BPMN Modeling and Reference Guide: Understanding and Using BPMN ; Develop Rigorous yet Understandable Graphical Representations of Business Processes*. Lighthouse Point, Fla: Future Strategies Inc.

Whitfield, R.I., A.H.B. Duffy, P. York, D. Vassalos, and P. Kaklis. 2011. “Managing the Exchange of Engineering Product Data to Support through Life Ship Design.” *Computer-Aided Design* 43 (5): 516–32. https://doi.org/10.1016/j.cad.2010.12.002.

Willner, Olga, Jonathan Gosling, and Paul Schönsleben. 2016. “Establishing a Maturity Model for Design Automation in Sales-Delivery Processes of ETO Products.” *Computers in Industry* 82 (October): 57–68. https://doi.org/10.1016/j.compind.2016.05.003.

Witherell, Paul, Sundar Krishnamurty, and Ian R. Grosse. 2007. “Ontologies for Supporting Engineering Design Optimization.” *Journal of Computing and Information Science in Engineering* 7 (2): 141. https://doi.org/10.1115/1.2720882.

Wölkl, Stefan. 2012. “Model Libraries for Conceptual Design.” Dissertation, München: TU München.

Wu, Zhaohong, Matthew I. Campbell, and Benito R. Fernández. 2008. “Bond Graph Based Automated Modeling for Computer-Aided Design of Dynamic Systems.” *Journal of Mechanical Design* 130 (4): 041102. https://doi.org/10.1115/1.2885180.

Wyatt, David F., David C. Wynn, Jerome P. Jarrett, and P. John Clarkson. 2012. “Supporting Product Architecture Design Using Computational Design Synthesis with Network Structure Constraints.” *Research in Engineering Design* 23 (1): 17–52. https://doi.org/10.1007/s00163-011-0112-y.

Wynn, David C., and P. John Clarkson. 2018. “Process Models in Design and Development.” *Research in Engineering Design*. https://doi.org/10.1007/s00163-017-0262-7.

Wynn, David C., and Claudia M. Eckert. 2016. “Perspectives on Iteration in Design and Development.” *Research in Engineering Design*, 1–32.

Yang, Kai, and Basem S. El-Haik. 2003. *Design for Six Sigma*. McGraw-Hill New York.

Yin, Su, and Jonathan Cagan. 2000. “An Extended Pattern Search Algorithm for Three-Dimensional Component Layout.” *Journal of Mechanical Design* 122 (1): 102. https://doi.org/10.1115/1.533550.

Yoshioka, Masaharu, Yasushi Umeda, Hideaki Takeda, Yoshiki Shimomura, Yutaka Nomaguchi, and Tetsuo Tomiyama. 2004. “Physical Concept Ontology for the Knowledge Intensive Engineering Framework.” *Advanced Engineering Informatics*, Ontology and it’s Applications to Knowledge-Intensive Engineering, 18 (2): 95–113. https://doi.org/10.1016/j.aei.2004.09.004.

You, Chun-Fong, Yu-Hsuan Yang, and Da-Kun Wang. 2011. “Knowledge-Based Engineering Supporting Die Face Design of Automotive Panels.” In *Industrial Design - New Frontiers*, edited by Denis Coelho. InTech.

Yu, Jiaqing, Jianzhong Cha, and Yiping Lu. 2012. “Design Synthesis Approach Based on Process Decomposition to Design Reuse.” *Journal of Engineering Design* 23 (7): 526–43. https://doi.org/10.1080/09544828.2011.629316.

Zachman, J. A. 1987. “A Framework for Information Systems Architecture.” *IBM Systems Journal* 26 (3): 276–92. https://doi.org/10.1147/sj.263.0276.

Zeiss, Benjamin, Diana Vega, Ina Schieferdecker, Helmut Neukirchen, and Jens Grabowski. 2007. “Applying the Iso 9126 Quality Model to Test Specifications.” *Software Engineering* 15 (6): 231–242.

Zhang, Linda L. 2014. “Product Configuration: A Review of the State-of-the-Art and Future Research.” *International Journal of Production Research* 52 (21): 6381–98. https://doi.org/10.1080/00207543.2014.942012.

Zimmermann, Luca, Tian Chen, and Kristina Shea. 2018. “A 3D, Performance-Driven Generative Design Framework: Automating the Link from a 3D Spatial Grammar Interpreter to Structural Finite Element Analysis and Stochastic Optimization.” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 32 (02): 189–99. https://doi.org/10.1017/S0890060417000324.

Appendix

In the following, the yielded results of the methods for identification of design automation use cases and derivation of metrics are listed.

Case 1 – Structural Analysis

See Figure 6-9.

Case 2 – Hydraulic Systems

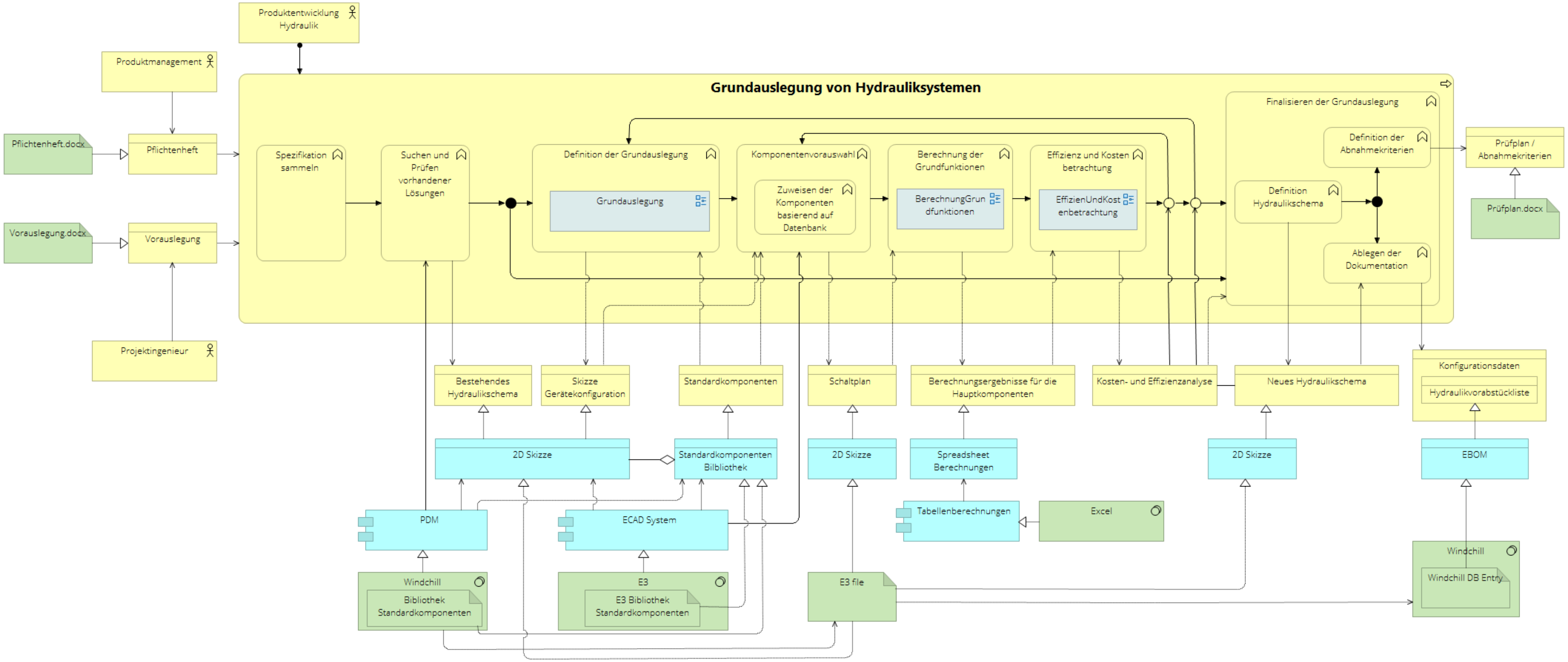


Figure 0-1: Micro-level design process for Case 2 – Hydraulic Systems. Blue boxes in the business layer (yellow) refer to links to detailed views shown in Figure 0-2, Figure 0-3 and Figure 0-4

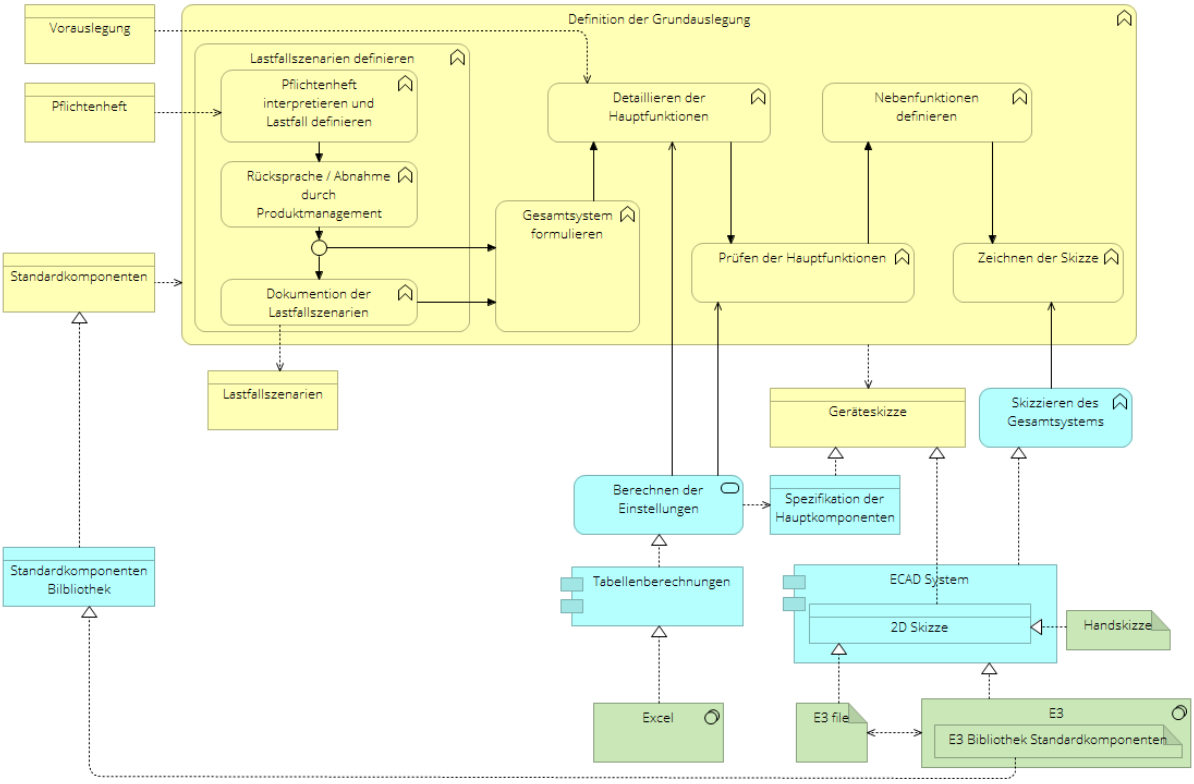


Figure 0-2: Micro-level design process for definition of the initial hydraulic layout (Case 2)

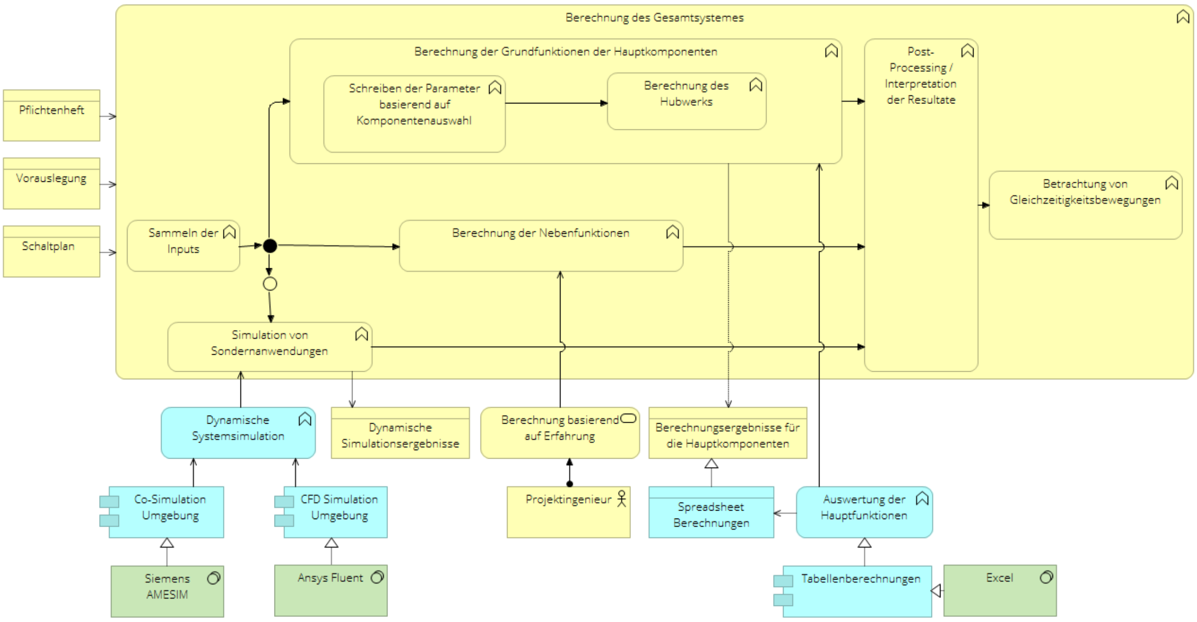


Figure 0-3: Micro-level design process for analysis of the hydraulic layout (Case 2)

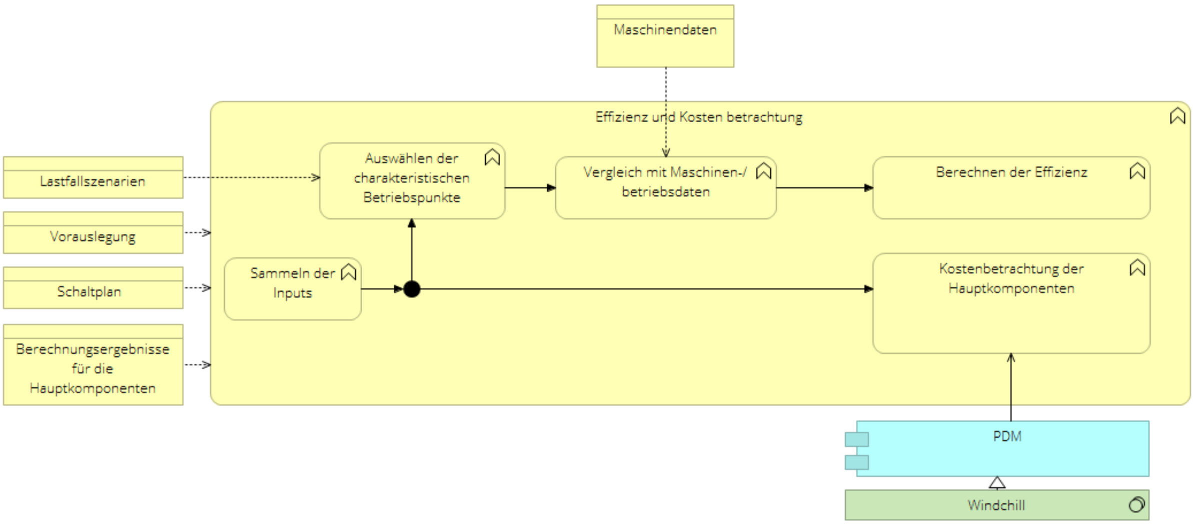


Figure 0-4: Micro-level design process for evaluation of efficiency and cost of hydraulic system (Case 2)

Case 3 – Hydraulic Units

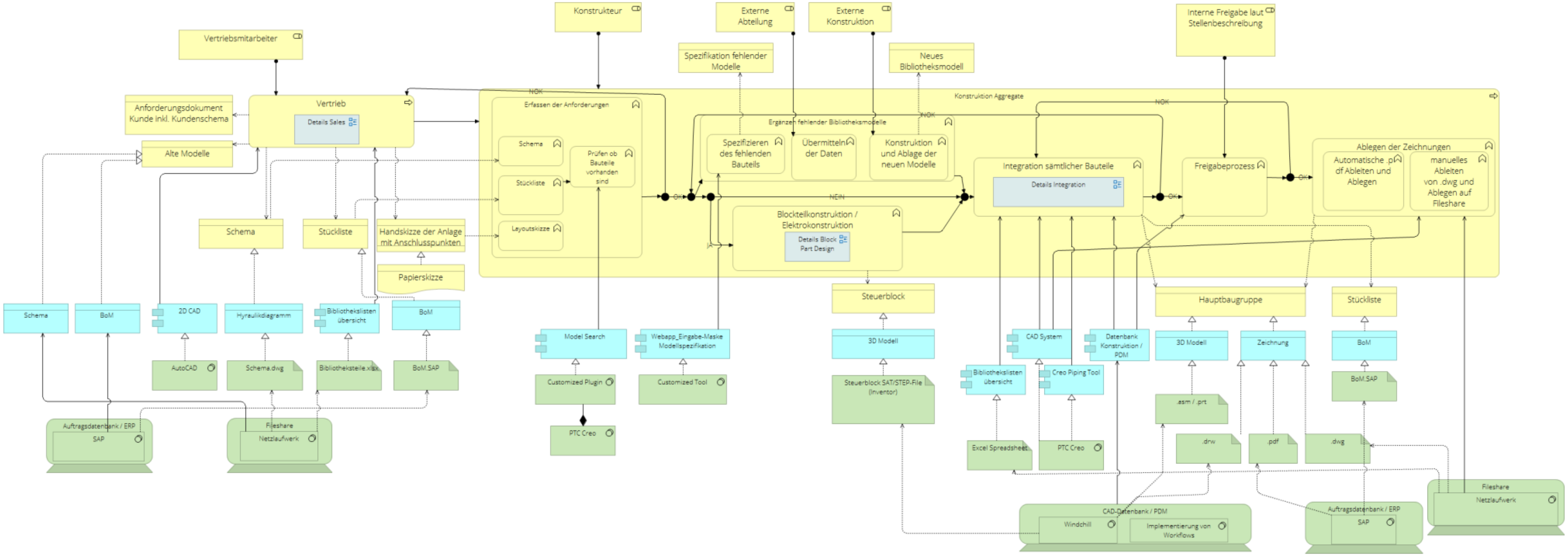
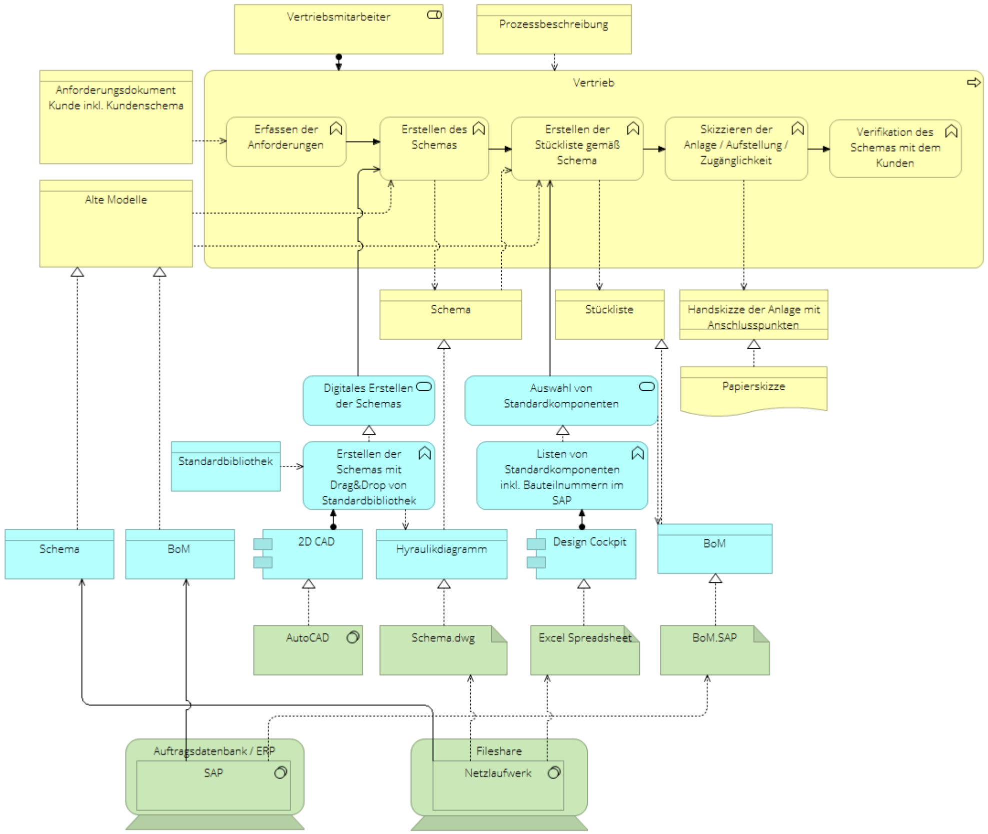


Figure 0-5: Micro-level design process for Case 3 – Hydraulic Aggregates. Blue boxes in the business layer (yellow) refer to links to detailed views shown in Figure 0-6, Figure 0-7 and Figure 0-8.



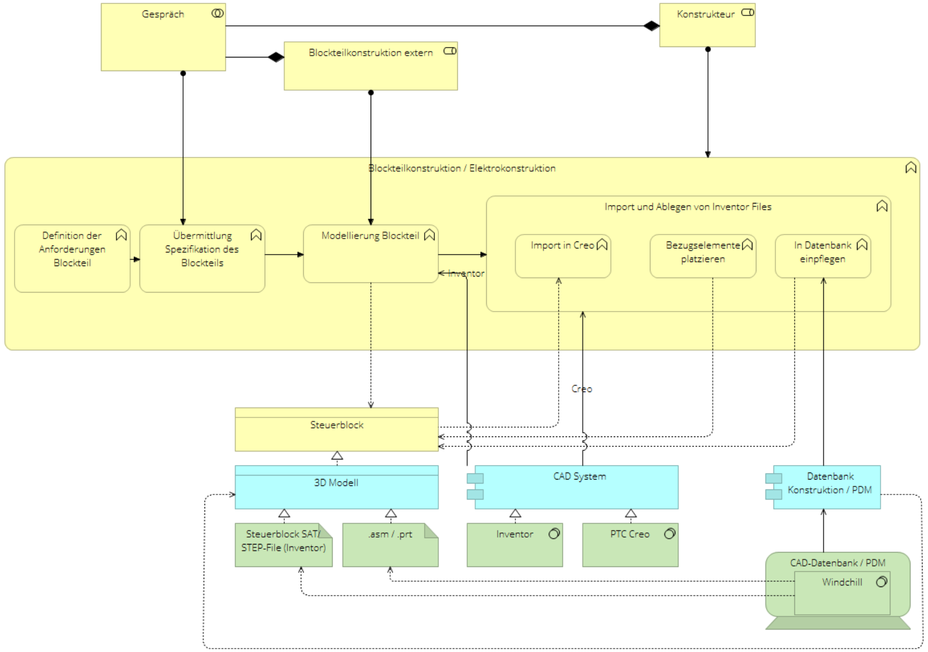


Figure 0-6: Micro-level design process for sales (Case 3).

Figure 0-7: Micro-level design process for block part design (Case 3).

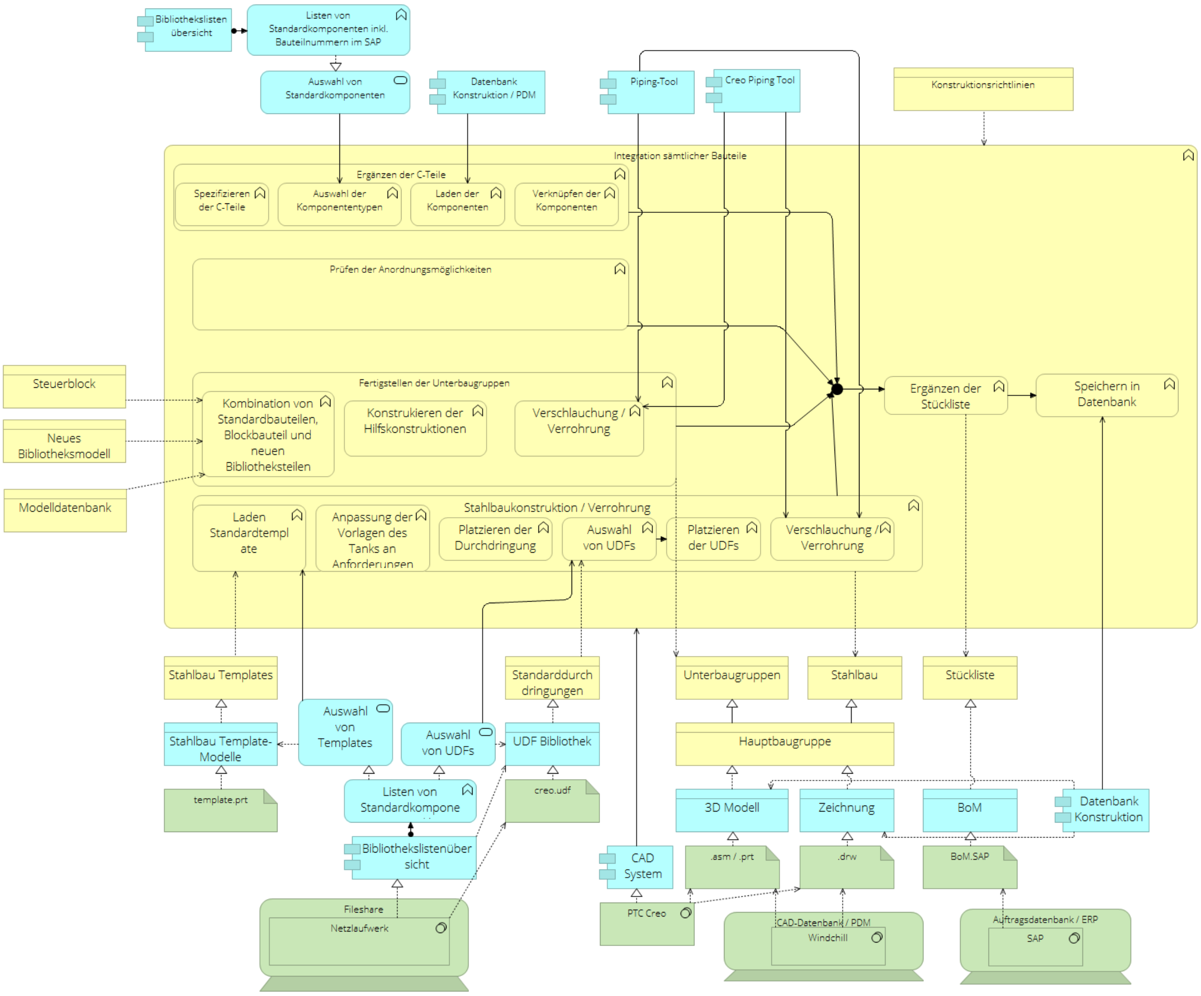


Figure 0-8: Micro-level design process for integration of functional components (Case 3)

Appendix A.2 Instantiated Task Templates

In the following, the instantiated task templates that account for the design process perspective are implemented for the three industrial cases.

Case 1 – Structural Analysis

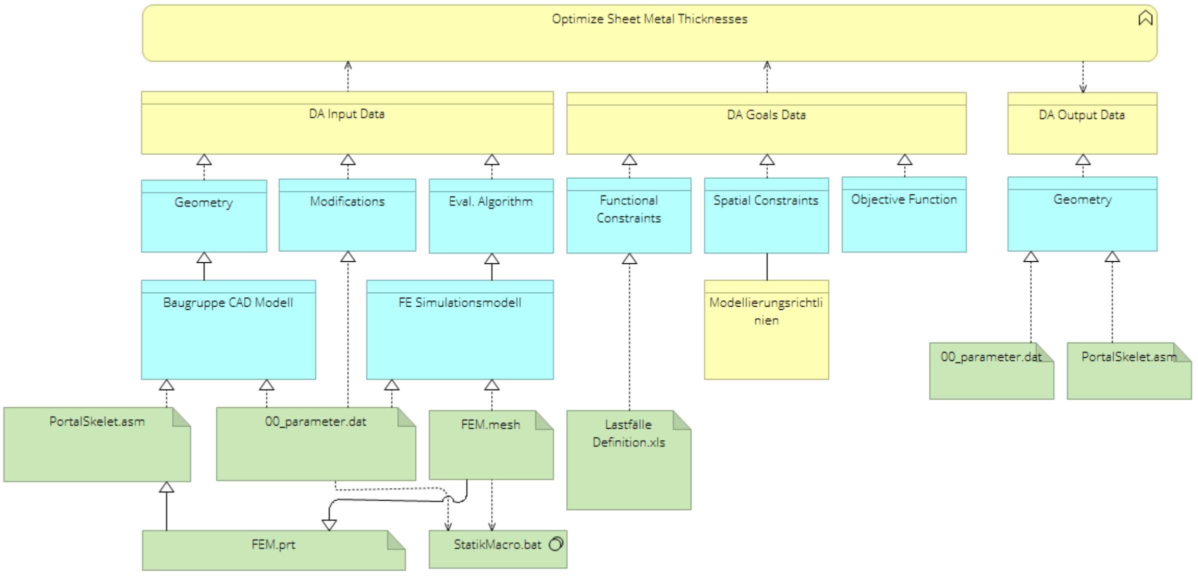


Figure 0-9: Instantiated SPAPS template for automated sizing of sheet metal thicknesses incl. mass optimization (Case 1)

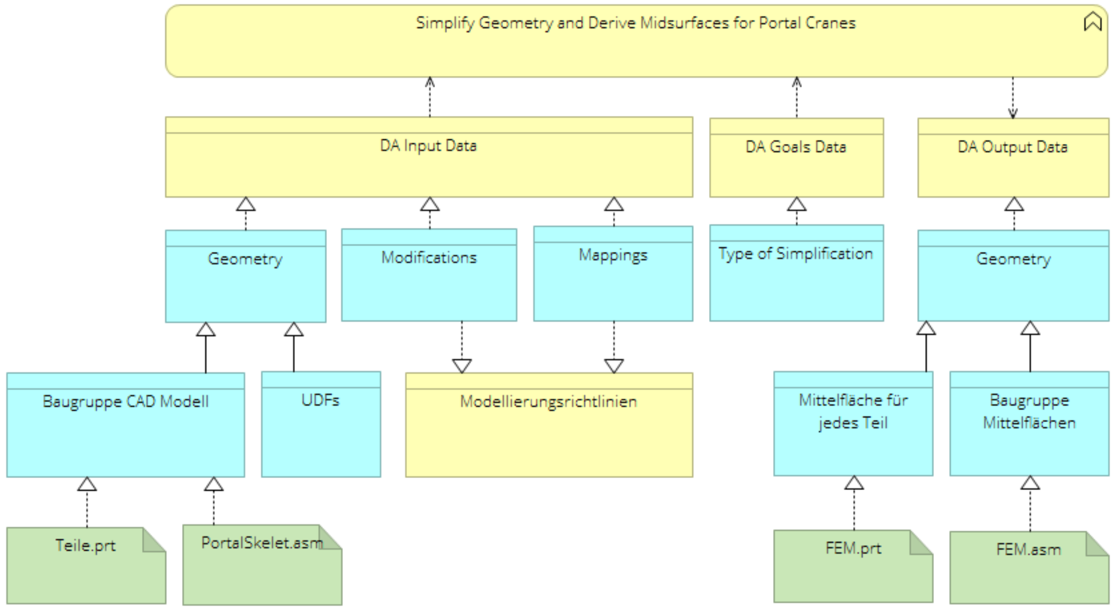


Figure 0-10: Instantiated SPAP template for automated midsurface generation for simulations (Case 1)

Case 2 – Hydraulic Systems

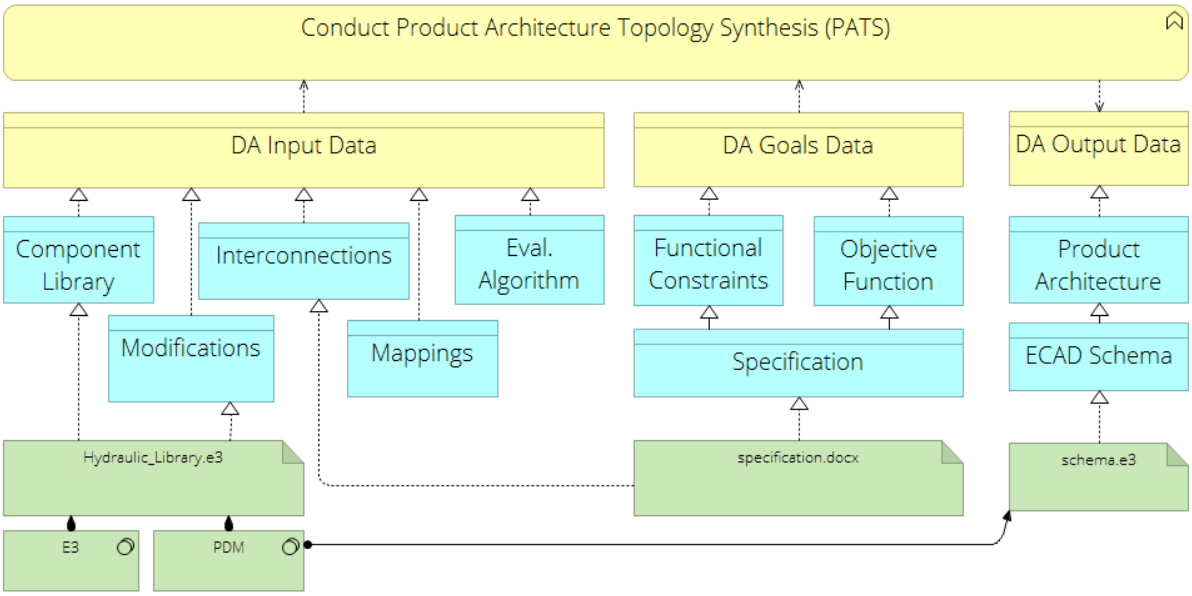


Figure 0-11: Instantiated PATS template for automated generation of hydraulic circuits (Case 2)

The instantiated PAPS template for automated component selection for closed circuits is depicted in Figure 6-10.

Case 3 – Hydraulic Units

Regarding the identification of use cases for design of hydraulic aggregates, multiple use cases could be identified based on the listing of use cases in Table 5-1 even though the majority of data for design automation implementation is missing for most of the use cases. Hence, the implementation of the design automation task template is implemented solely for the case of design configuration to support 2D configuration of hydraulic aggregates in sales, see Figure 0-12.

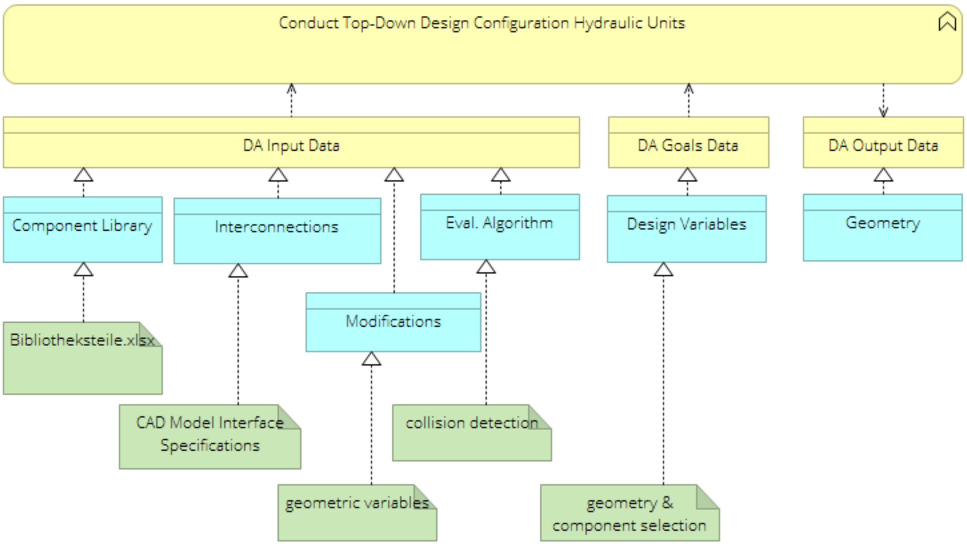


Figure 0-12: Instantiated design configuration template for 2D configuration of hydraulic units in sales (Case 3).

Appendix A.3 Micro-Level Processes with Design Automation

Case 1 – Structural Analysis

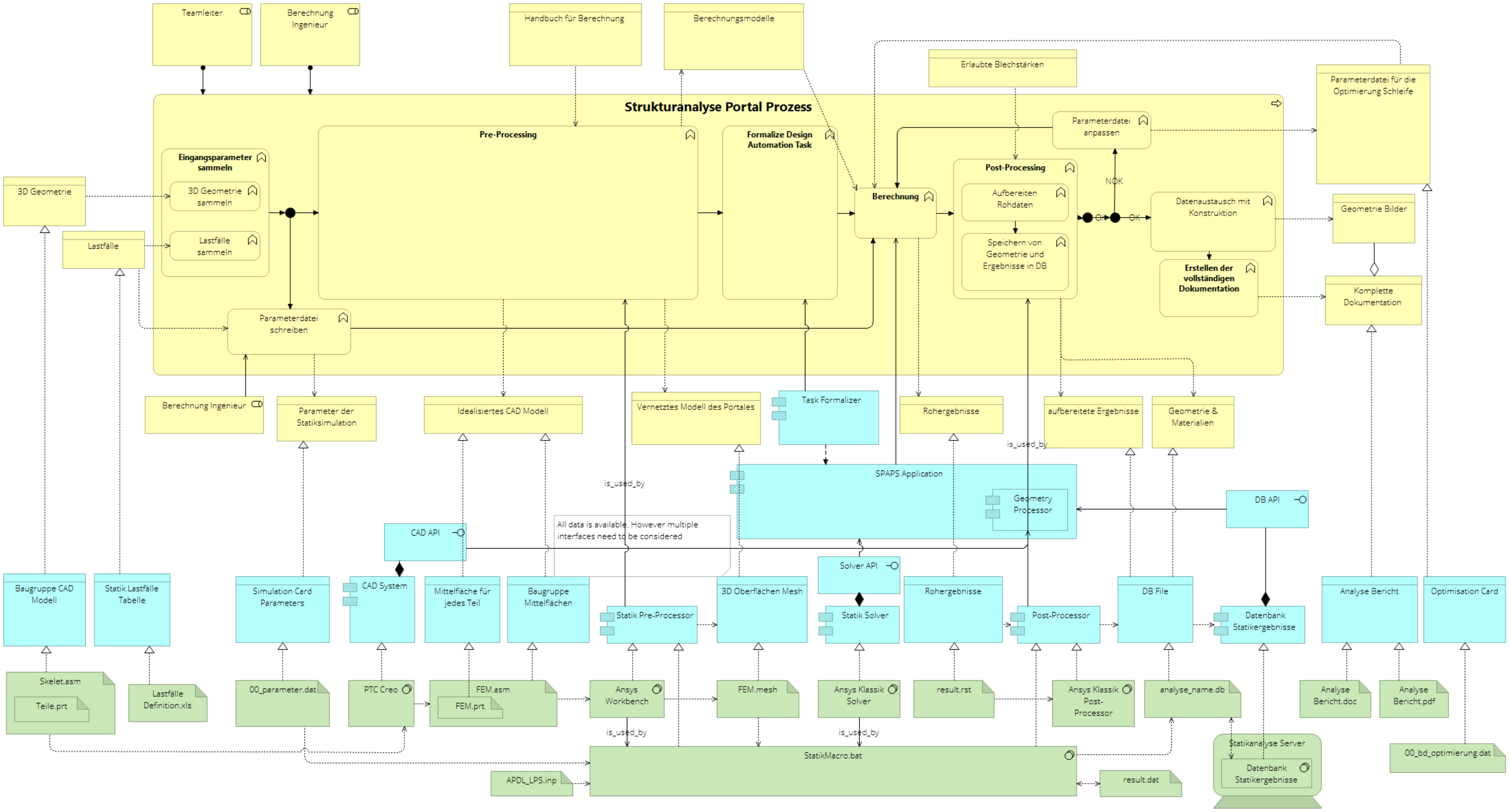


Figure 0-13: Micro-level process with integrated SPAPS template for automated sizing of sheet metal thicknesses incl. mass optimization (Case 1).

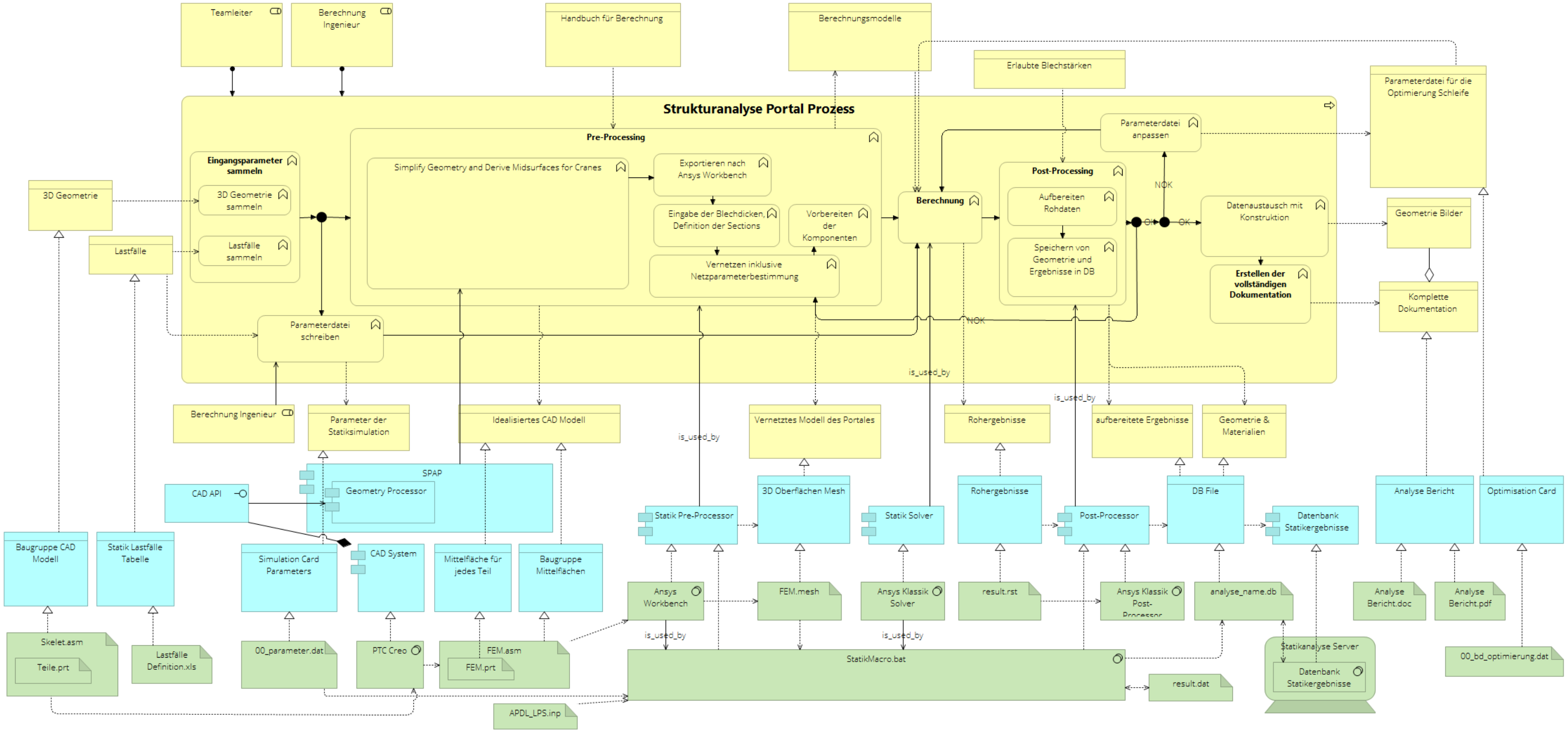


Figure 0-14: Micro-level process with integrated SPAP for automated midsurface generation for simulations (Case 1)

Case 2 – Hydraulic Systems

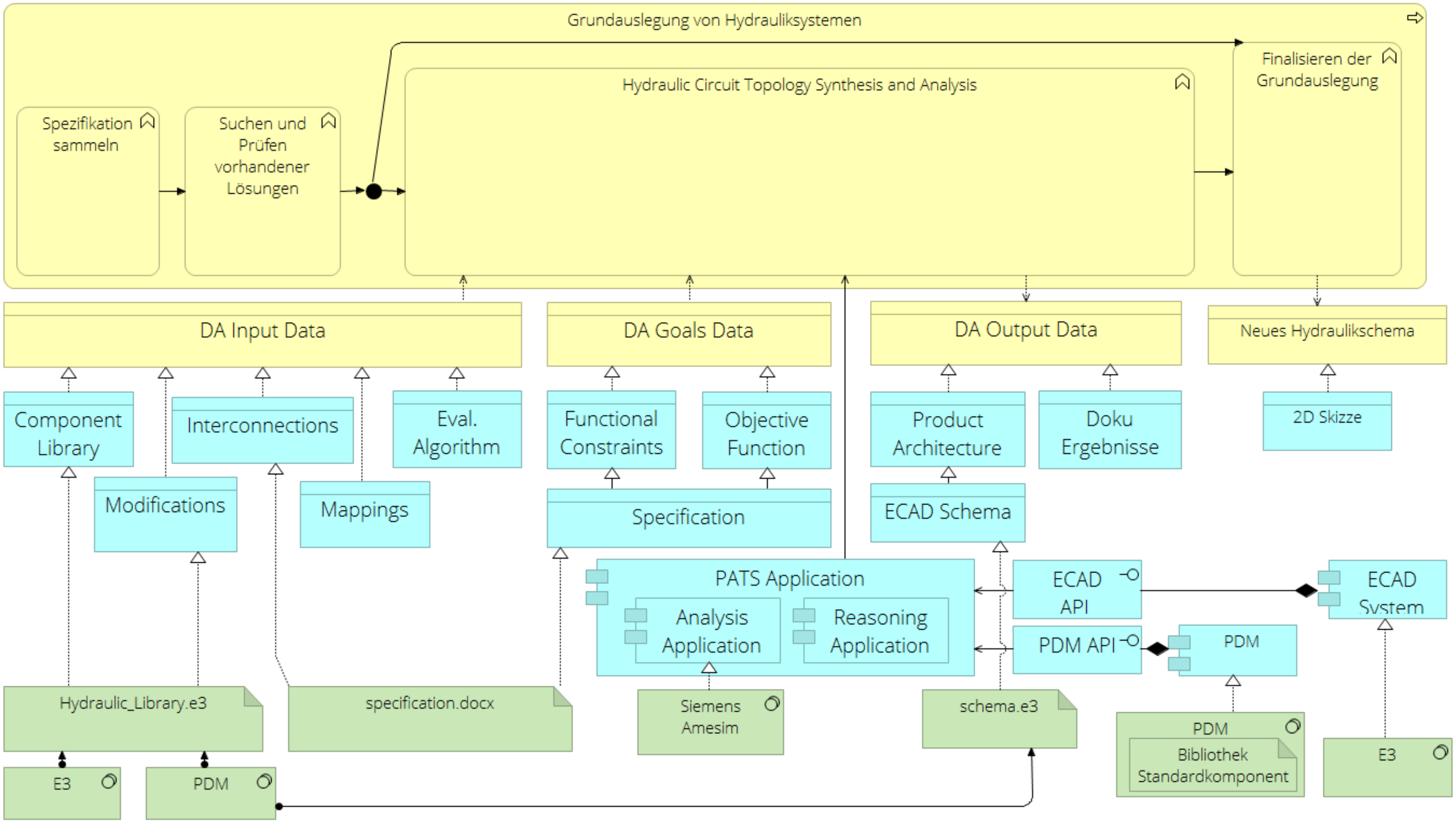


Figure 0-15: Micro-level process with integrated PAPS template for automated selection of hydraulic component for closed circuits (Case 2)

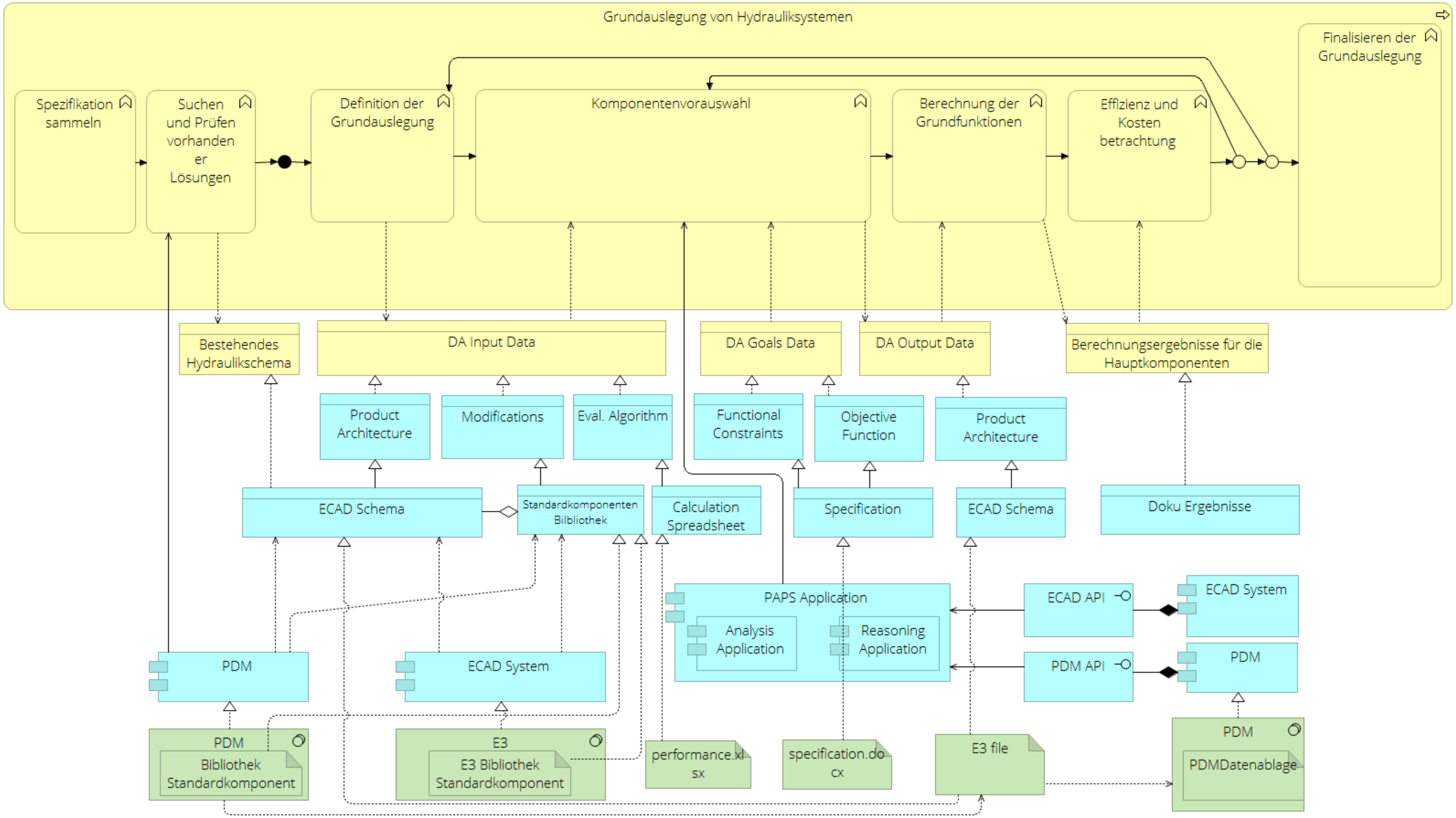


Figure 0-16: Micro-level process with integrated PATS template for automated generation of hydraulic circuit concepts (Case 2)

Case 3 – Hydraulic Units

For the case of hydraulic units, the modeling of potential future scenarios is considered obsolete since the majority of information is currently missing / not available in a formalized format. Instead, the potential design automation application is integrated to the design automation task template to highlight the expected interfaces of the application, see Figure 0-17.

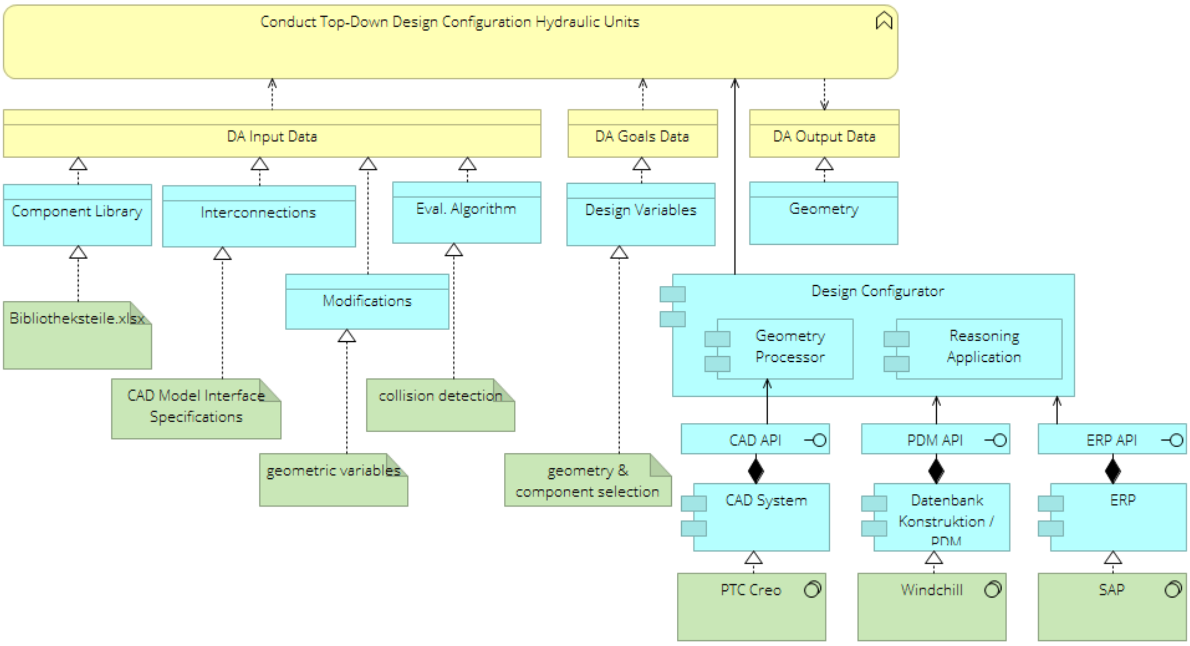


Figure 0-17: Micro-level process with integrated design configuration template for 2D configuration in sales (Case 3).

Appendix A.4 Value Stream Mapping

Case 1 – Structural Analysis

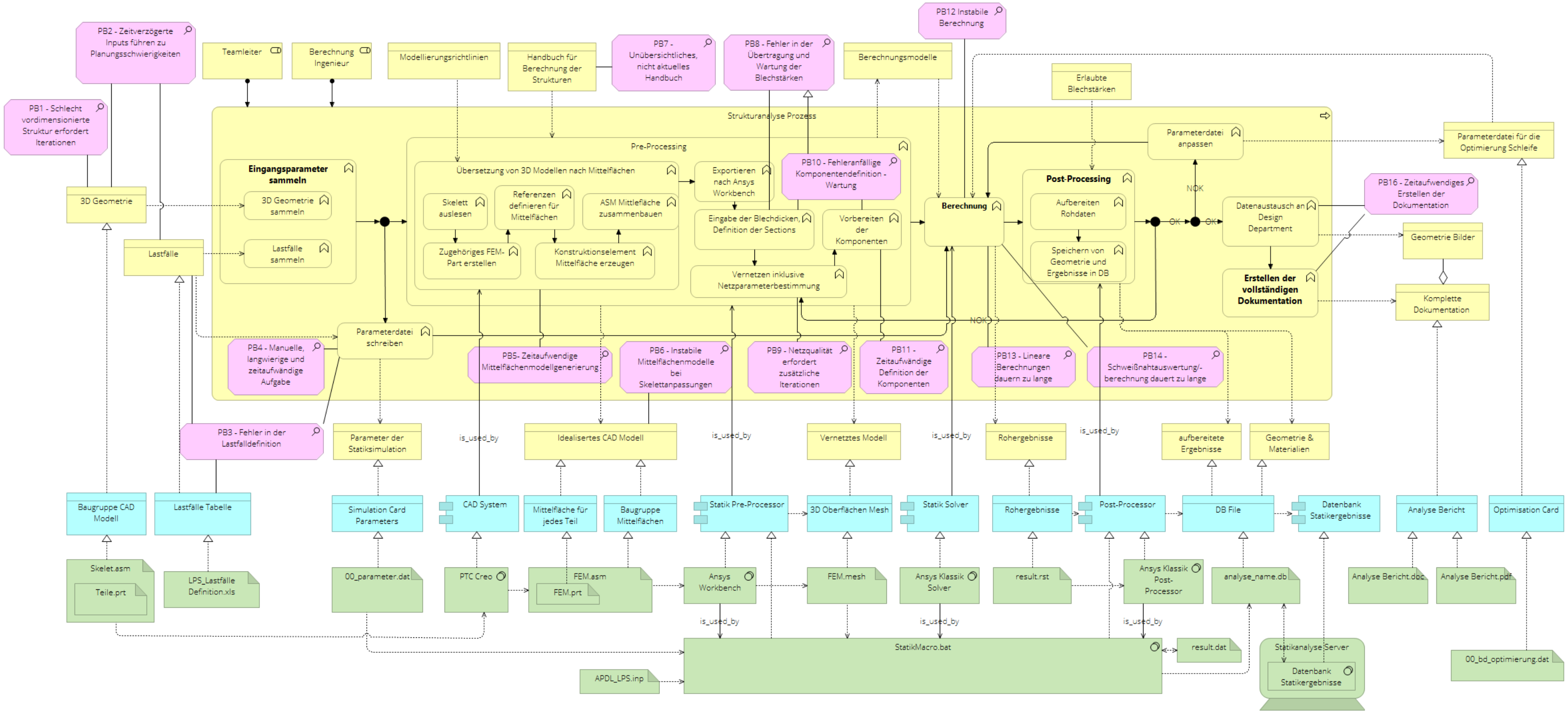


Figure 0-18: Values stream mapping for Case 1 – Structural Analysis. Identified failure modes are highlighted in purple boxes.

Case 2 – Hydraulic Systems

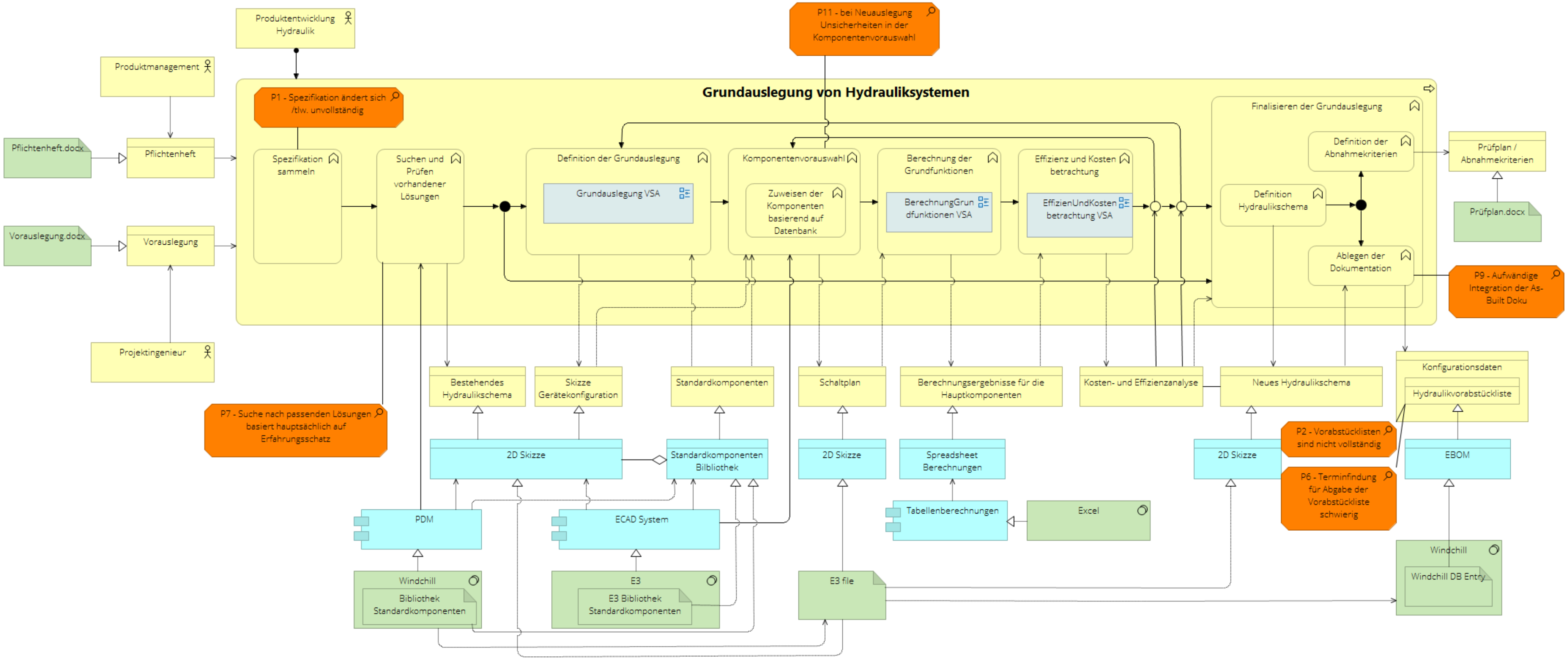


Figure 0-19: Values stream mapping for Case 2 – Hydraulic Systems. Blue boxes in the business layer (yellow) refer to links to detailed views shown in Figure 0-20, Figure 0-21 and Figure 0-22. Identified failure modes are highlighted in orange boxes.

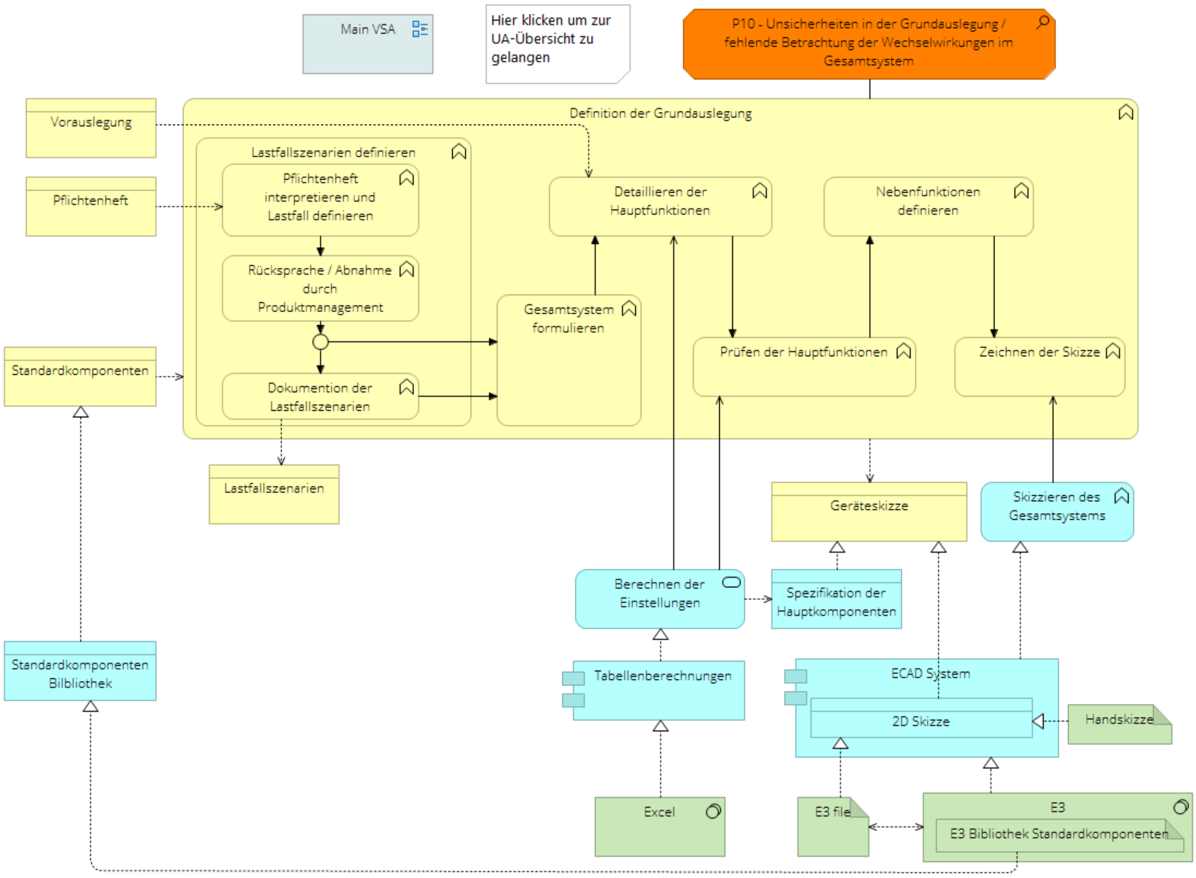


Figure 0-20: Value stream mapping for definition of initial layout (Case 2). Identified failure modes are highlighted in orange boxes.

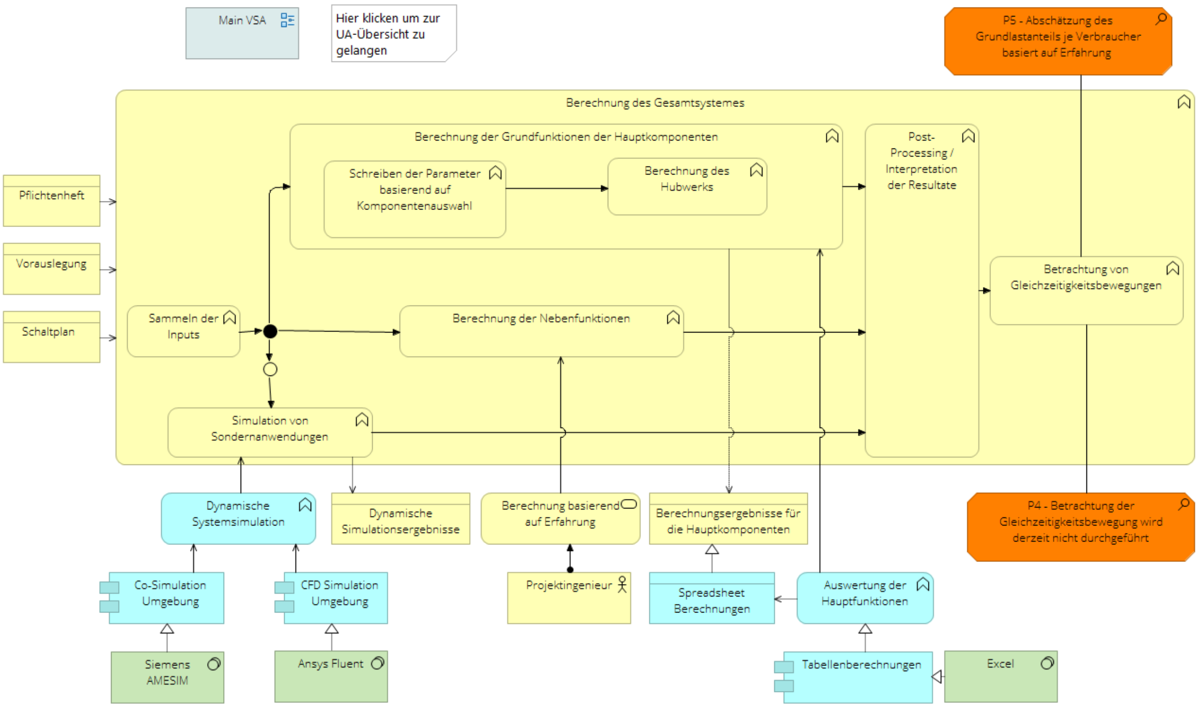


Figure 0-21: Value stream mapping for analysis of layout (Case 2). Identified failure modes are highlighted in orange boxes.

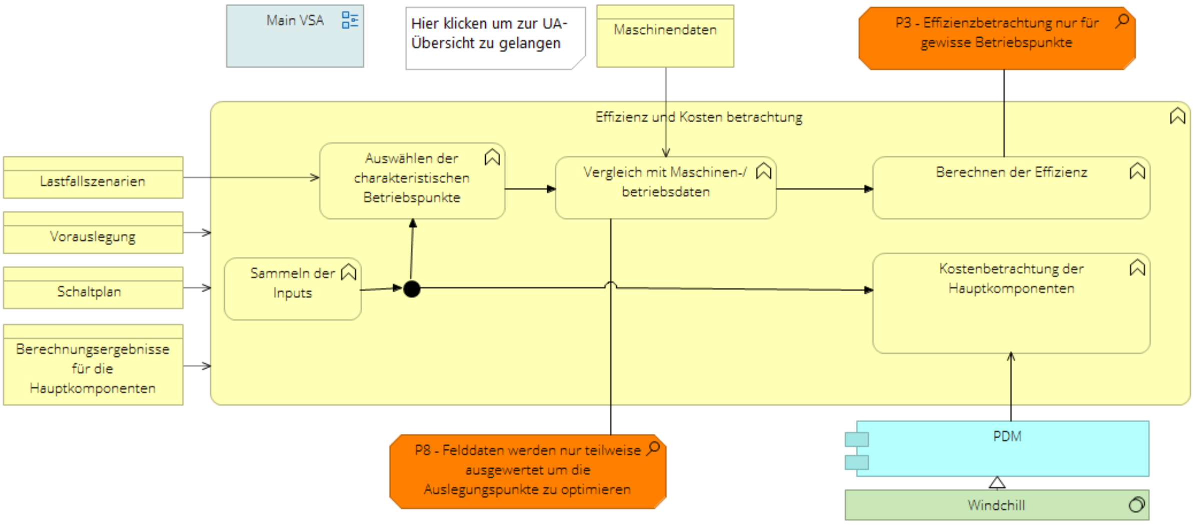


Figure 0-22: Value stream mapping for evaluation of efficiency and cost of layout (Case 2). Identified failure modes are highlighted in orange boxes.

Appendix A.5 Failure Mode Effect Analysis

Case 1 – Structural Analysis

Table 0-1: Problem cause effects analysis based on FMEA for Case 1 – Structural Analysis.

Prozessschritt / Aktivität	Probleme ("Failure Mode")	Effekte durch Auftreten der Probleme (wie wirkt sich das Problem aus)	Ursachen für das Auftreten der Probleme
Eingangsparameter sammeln und Parameterdatei schreiben	**PB1 - Schlecht vordimensionierte Struktur**	Arbeit dauert länger und ist ressourcenintensiver aufgrund zusätzlicher Iterationen. Instabile Berechnungen sind möglich. Durch schlecht abgeschätzte Systemmaße wird das Optimum trotz vieler Iterationen nicht erreicht und das Struktur	Zeitdruck in der Konstruktionsabteilung
			Fehlende Erfahrung des Konstrukteurs
			Zuverlässigkeit der Vorberechnung
			Kein Kontrollmechanismus vor Übergabe an Statik
			Konstruktion kann nicht gefertig werden
			Kundenwüsche ändern sich
	PB2 - Verspätete Inputs Übergabedokumenten	Wartezeiten und mögliche Verzögerungen in der Ergebnislieferung	Ändernde Kundenspezifikation
			Projektgeschäft schwierig zu planen (Über- bzw. Unterkapazitäten)
	PB3 - Fehler in der Lastfalldefinition	Instabile Berechnungen, Iterationen mit der Konstruktionsabteilung und intern. Dadurch Verzögerungen in der Ergebnislieferung und Fehlauslegung möglich.	Konzentration des Konstrukteur
			Kein Kontrollmechanismus für Lastfalldatei
			Kein Kontrollmechanismus vor Übergabe an Statik
			Nur manuelle Übertragung
	PB4 - Manuelle, langwierige und zeitaufwändige Aufgabe	Motivation der Mitarbeiter leidet, Iterationen aufgrund von Fehlern	Konzentration des Konstrukteur
			Kein Kontrollmechanismus für Lastfalldatei
			Kein Kontrollmechanismus vor Übergabe an Statik
			Nur manuelle Übertragung
Pre-Processing	**PB5- Zeitaufwendige Mittelflächenmodellgenerierung**	Keine automatische Ableitung von Mittelflächen vorhanden. Jede Mittelfläche wird auf Basis des Skels konstruiert. Blechdickensprünge müssen vermieden werden. Somit manuelle Eingabe von Blechdicken und Offsets erforderlich. Modellierung von Blechdickensprüngen.	Zeitaufwendig aufgrund vieler
			Modellierungsfehler in 3D Modell von Konstruktion
			Kein Unterstützungswerkzeug
			Kein Kontrollmechanismus bei Übergabe an die Statik
			Aufwendiger Prüfmechanismus
			Standard für die Modellierung deckt nicht sämtliche Varianten ab
			je komplexer das Design, desto aufwändiger die Generierung des Mittelflächenmodelles

Prozessschritt / Aktivität	Probleme ("Failure Mode")	Effekte durch Auftreten der Probleme (wie wirkt sich das Problem aus)	Ursachen für das Auftreten der Probleme
Pre-Processing	**PB6 - Instabile Mittelflächenmodelle bei Skelettanpassungen**	Änderungen des Skels werden nach einchecken des Skels in der tbf.asm sichtbar. Die Regeneration verläuft in den seltensten Fällen ohne eine Reparatur. Im schlimmsten Fall muss ein erheblicher Zeitaufwand für die Reparatur des Mittelflächenmodells erbracht werden.	Fehler in der Modellierung - Skeletmodell
			Fehler in der Modellierung - Mittleflächenmodell
			Modellierungsmethoden werden konstruktionsseitig nicht eingehalten?
			Unachtsamkeiten im Prüfprozess der Mittelflächemodelle
			Modellierungsrichtlinien in der Statik werden nicht eingehalten
			Modellierung in der Konstruktionsabteilung basiert auf Erfahrungswerten
			Je nach Modellierung muss die Ableitung der Mittelflächenmodelle angepasst werden.
			Fehlendes Bewusstsein für Einfluss von Änderungen bzw. Anpassungen der Geometrie auf Mittelflächenmodell
			Modellierungsrichtlinien in der Statik werden nicht eingehalten
			Manueller Ablauf in Creo
	PB7 - Unübersichtliches, nicht aktuelles Handbuch	Der Unerfahrene Bearbeiter macht Fehler oder muss Rücksprache mit den Strukturverantwortlichen halten. Dies ist vor allem im Pre-Processing bei der Erstellung von Komponenten der Fall.	Fehlende Zeit für die Wartung
			Schwierigkeiten beim Formulieren für den Nutzer - Was sind wirklich die Probleme der Anwender?
			Roter Faden im Handbuch fehlt
			Fehlende Produktstandards fordern zusätzliche Erklärungen
			Unpassendes Medium für die Dokumentation
			Komplexe Navigation in der Dokumentation
			Handbuch ist Dokument-Basiert
			Unklare Prozesse erschweren Formulierung / Strukturrierung der Richtlinien
	PB8 - Fehler in der Übertragung und Wartung der Blechstärken	Blechstärken und Offset müssen für jede Mittelfläche von Hand eingegeben werden. Während der Optimierung müssen Änderungen vorgenommen werden. Fehler kann zu Fehlauslegung führen	Konzentration des Konstrukteur
			Fehlerhafte Blechdicken im Skeletmodell
			Keine vollumfängliche Kontrolle der Parameterdefinition (nur manuell)
			Viele händisch durchzuführende Schritte
			Keine Unterstützungssoftware für die Übertragung der Parameter - Manuelles Messen im Skelettmodell

Prozessschritt / Aktivität	**Probleme ("Failure Mode")**	**Effekte durch Auftreten der Probleme (wie wirkt sich das Problem aus)**	**Ursachen für das Auftreten der Probleme**
Pre-Processing	**PB9 - Netzqualität erfordert zusätzliche Iterationen**	Beieinflussung der Berechnungsergebnisse durch die Vernetzung. Problemstellen werden nicht erkant oder erst zu einem späteren Zeitpunkt sichtbar. Abweichungen zwischen Simulation und Realität was zu einer nichtkonservativen Auslegung führen kann.	Unzureichnende bzw. keine Kontrollmechanismen zur Bewertung
			Unzureichnende keine Kontrollmechanismus zur Bewertung der Netzqualität - Interne Bewertung prüft nur die vollständige Berechnung - daher zusätzliche Iterationen
			Fehlende regeln im Handbuch für die Erstellung des Netzes
			Fehlende Erfahrung des Ingenieurs
			Zeitmangel für Vernetzung
			Workbench Kontrolle wird nicht immer genutzt
			Komplex Geometrien erforden aufwändige vernetzung
	PB10 - Fehleranfällige Komponentendefinition - Wartung		Selbe Ursache wie PB8. Daher wurde PB8 um "und Wartung" ergänzt und die Probleme zusammengeführt
	PB11 - Zeitaufwändige Definition der Komponenten	Fehler in der Komponentenerstellung führen meistens zu Fehlern und zum Abbruch der Berechnung.	Strukturinput in Ansys Workbench sollte die Definition der Komponenten beinhalten
			Keine automatisierte Ableitung der Komponenten
			Ingenieur Konzentration
			Viele manuelle Arbeitsschritte
Berechnung	**PB12 - Instabile Berechnung**	Zeitaufwände für die Identifizierung von Gründen für die Instabilität; Iterationen im Prozess; mögliche Verzögerungen in der Ergebnislieferung	Fehlerhafte Modelldefinition - Netz
			Fehlerhafte Modelldefinition - Lastfälle
			Fehlerhafte Modelldefinition - Blechstärken
			Fehler im Input der Konstruktionsabteilung - Geometrie
			Fehler im Input der Konstruktionsabteilung - Lastfälle
			Fehler im Input der Konstruktionsabteilung - Netz
			Komplexe / nicht-standard Geometrien
			Schlecht vordimensionierte Geometrien
			Komplexe Anleitungen verkomplizieren Aufbereitung der Daten
			Keine Fehlermeldung bei Absturz bei Berechnung der Strukturinputs (Lastfälle, Aufbringen der Lastfälle, Aufbereiten der Ergebnisse etc.)
	PB13 - Lineare Berechnungen dauern zu lange	Durch eine lange Rechenzeit wird der Bearbeitungszeitraum für die Optimierung vergrößert.	Angst vor neuen Methoden
			Keine Optimierung im Rechnungsprozess
			Unpassende Methoden in den Berechnungen
	PB14 - Schweißnahtauswertung/-berechnung dauert zu lange	Langes Post-processing in jeder Optimierungsschleife.	Angst vor neuen Methoden - Verlust von Einfluss
			Veraltete Methoden

Prozessschritt / Aktivität	Probleme ("Failure Mode")	Effekte durch Auftreten der Probleme (wie wirkt sich das Problem aus)	Ursachen für das Auftreten der Probleme
Post-processing	**PB15 - Optimierung Personenressourcenintensiv**	Iteratives Vorgehen bei Bauteiloptimierung (konstruktive Aspekte müssen Beachtung finden). Erfolg der Optimierung hängt stark von der Erfahrung des	Lange Wartezeiten für die Simulation
			Fehlende Erfahrung / Verständnis für die Struktur
	PB16 - Zeitaufwendiges Erstellen der Dokumentation	Die Dokuementation erfolgt meistens sehr spät nach der Bearbeitung. Ein Feststellen von Fehlern in dieser späten Phase kann mit hohen Folgekosten verbunden sein.	Veraltete Methoden
			Angst von Veränderungen
			Automatisierung stellt die Ergebnisse unverständlich dar, z.B. unpassende Zoomeinstellung
			Word erlaubt nur Teilautomatiserung
			Macro für die automatisierte Erstellung der Bilder fehlt

Case 2 – Hydraulic Systems

Table 0-2: Problem cause effects analysis based on FMEA for Case 2 – Hydraulic Systems.

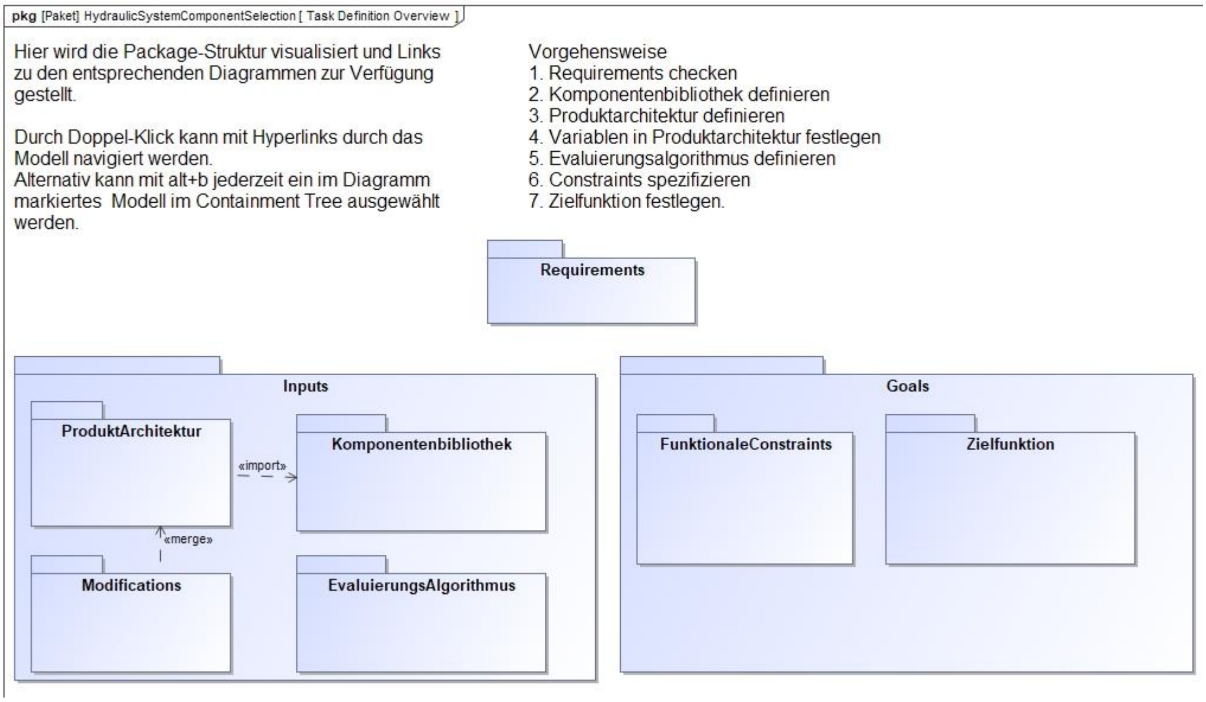
Prozessschritt / Aktivität	Probleme ("Failure Mode")	Effekte durch Auftreten der Probleme (wie wirkt sich das Problem aus)	Ursachen für das Auftreten der Probleme
Spezifikationen sammeln	**P1 - Spezifikation ändert sich /tlw. unvollständig**	Zusätzliche Iterationen in der Grundauslegung.	Kundenanwendungen variieren
			Änderungswünsche von Produktmanagement
Prüfen vorhandener Lösungen	**P7 - Suche nach passenden Lösungen basiert hauptsächlich auf Erfahrungsschatz**	Ingenieure entwickeln bestehende Systeme neu. Zusätzlicher Betreuungsaufwand.	Mangelnde Suchfunktionalität in der Datenbank
			Fehlende Gesamtübersicht bereits entwickelter Systeme
Definition der Grundauslegung	**P10 - Unsicherheiten in der Grundauslegung / fehlende Betrachtung der Wechselwirkungen im Gesamtsystem**	Versagen am Prüfstand. Neudefinition der Komponente, dadurch Zeitverluste. Zusätzliche Iterationen.	Unbekannte aufgrund Neuentwicklung
			Fehlende Erfahrung / Wissen
			Mangelnde technische Mittel (Simulation)
Komponentenvorauswahl	**P11 - bei Neuauslegung Unsicherheiten in der Komponentenvorauswahl**	Zusätzliche Iterationen in der Komponentenauswahl und Effizienzbetrachtung.	Fehlende Erfahrung / Wissen
Berechnung der Grundfunktionen	**P4 - Betrachtung der Gleichzeitigkeitsbewegung wird derzeit nicht durchgeführt**	Effizienzreduktion bei Gleichzeitigkeitsbewegungen (offener Kreis).	Unsicherheit / keine Vorgaben in der Lastfalldefinition
			nicht schriftlich festgehaltene Vorgaben
	P5 - Abschätzung des Grundlastanteils der Verbraucher basiert auf Erfahrung	Zu niedrig -> Grenzlastregelung greift oft ein; zu hoch -> hoher Spritverbrauch	Beruht nur teilweise auf Daten
			Daten müssen interpoliert werden
			Herstellerdaten mangelhaft / ungenau

Prozessschritt / Aktivität	Probleme ("Failure Mode")	Effekte durch Auftreten der Probleme (wie wirkt sich das Problem aus)	Ursachen für das Auftreten der Probleme
Effizienz und Kostenbetrachtung	**P8 - Felddaten werden nur teilweise ausgewertet um die Auslegungspunkte zu optimieren**	Überdimensionierung / zu konservative Auslegung. Schäden an Antrieben aufgrund von Überlastung.	Fehlende Daten bei Neuentwicklung
			Zeitaufwändig
			Vielfältigkeit der Anwendungen - welcher Lastfallzyklus ist repräsentativ?
			Schwierigkeiten in der Dateninterpretation
			Daten sind fehlerbehaftet
	P3 - Effizienzbetrachtungen nur für gewisse Betriebspunkte	Teillastgeschwindigkeiten stimmen nicht. Ungenauigkeiten in der Geschwindigkeitsangabe. Einfluss auf	Beruht nur teilweise auf Daten
			Daten müssen interpoliert werden
			Herstellerdaten mangelhaft / ungenau
Finalisieren der Grundauslegung	**P9 - Aufwändige Integration der As-Built Doku**	Zeitaufwand, zusätzliche Iterationen, Revisionen	Freiheitsgrade in der Vorauslegung: Detaillierung der Verrohrung passiert in der Fertigung
			Änderungen werden nicht in der Originaldoku dokumentiert
	P2 - Vorabstücklisten sind nicht vollständig	Zeitverzug in der Fertigung	Komponenten Vergessen
			Unsicherheiten bei Lieferzeiten
			Manuelle Prüfung mit Datenbank zeitaufwendig
	P6 - Terminfindung für Abgabe der Vorabstückliste schwierig	Zeitverzug in der Fertigung	Unsicherheiten bei Lieferzeiten
			Wartung der Datenbank mit aktuellen Lieferzeiten

Appendix A.6 SysML Models

In the following, the models related to the user study introduced in Section 9.2 are presented. First, the SysML models addressing scenario 1 that focuses on adaptation of a component selection task for hydraulic circuits are shown. Second, the solution to scenario 2 that addresses the knapsack optimization problem is presented. For each scenario, the overview of the model is presented (PKG) as well as the details of the nested views (BDD, PAR).

Scenario 1 – Hydraulic Component Selection



Hier wird die Package-Struktur visualisiert und Links zu den entsprechenden Diagrammen zur Verfügung gestellt.

Durch Doppel-Klick kann mit Hyperlinks durch das Modell navigiert werden.
Alternativ kann mit alt+b jederzeit ein im Diagramm markiertes Modell im Containment Tree ausgewählt werden.

Vorgehensweise
1. Requirements checken
2. Komponentenbibliothek definieren
3. Produktarchitektur definieren
4. Variablen in Produktarchitektur festlegen
5. Evaluierungsalgorithmus definieren
6. Constraints spezifizieren
7. Zielfunktion festlegen.

Figure 0-23: PKG for navigation within the design automation task formalization for automated hydraulic component selection, Scenario 2

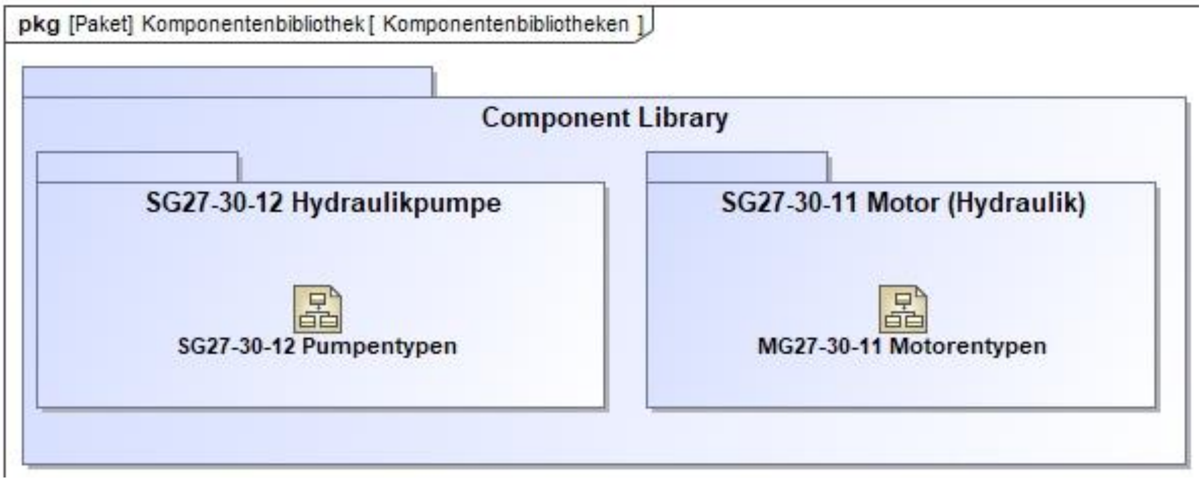


Figure 0-24: PKG for navigation within component library for Scenario 1

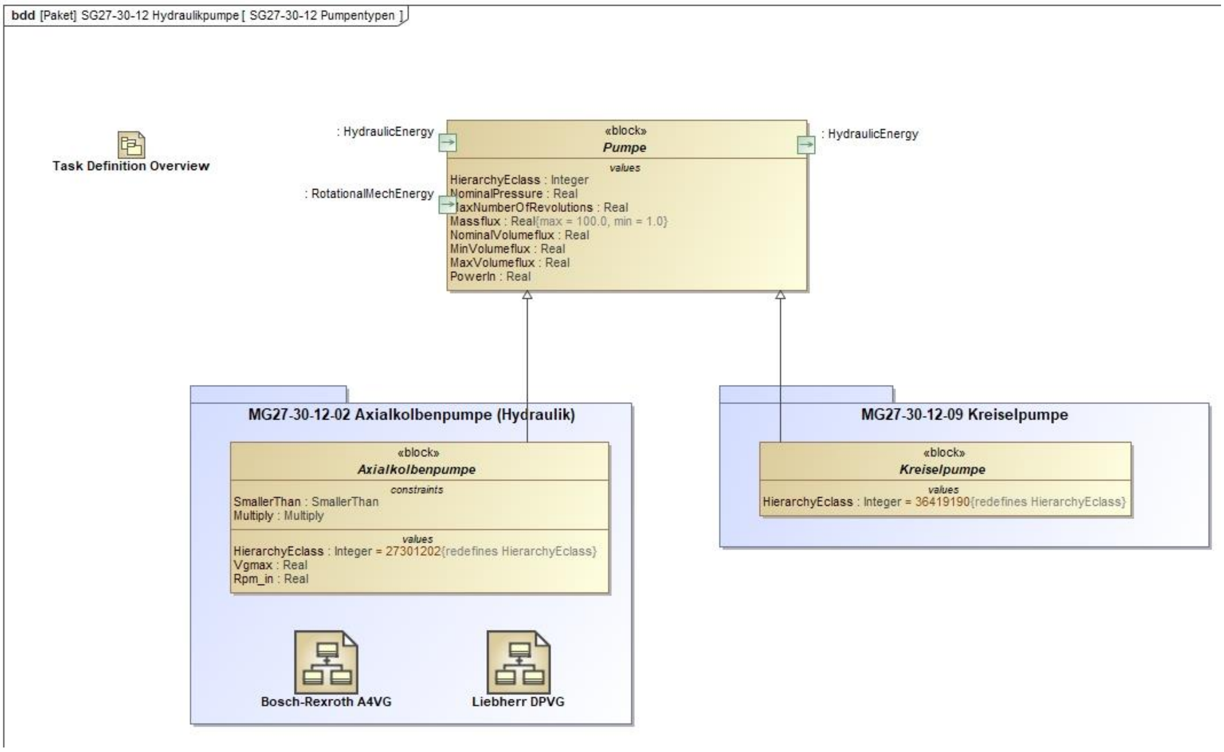


Figure 0-25: BDD showing the definition of abstract elements for different types of pumps, Scenario 1. The definition of libraries for hydraulic engines is conducted in a similar manner.

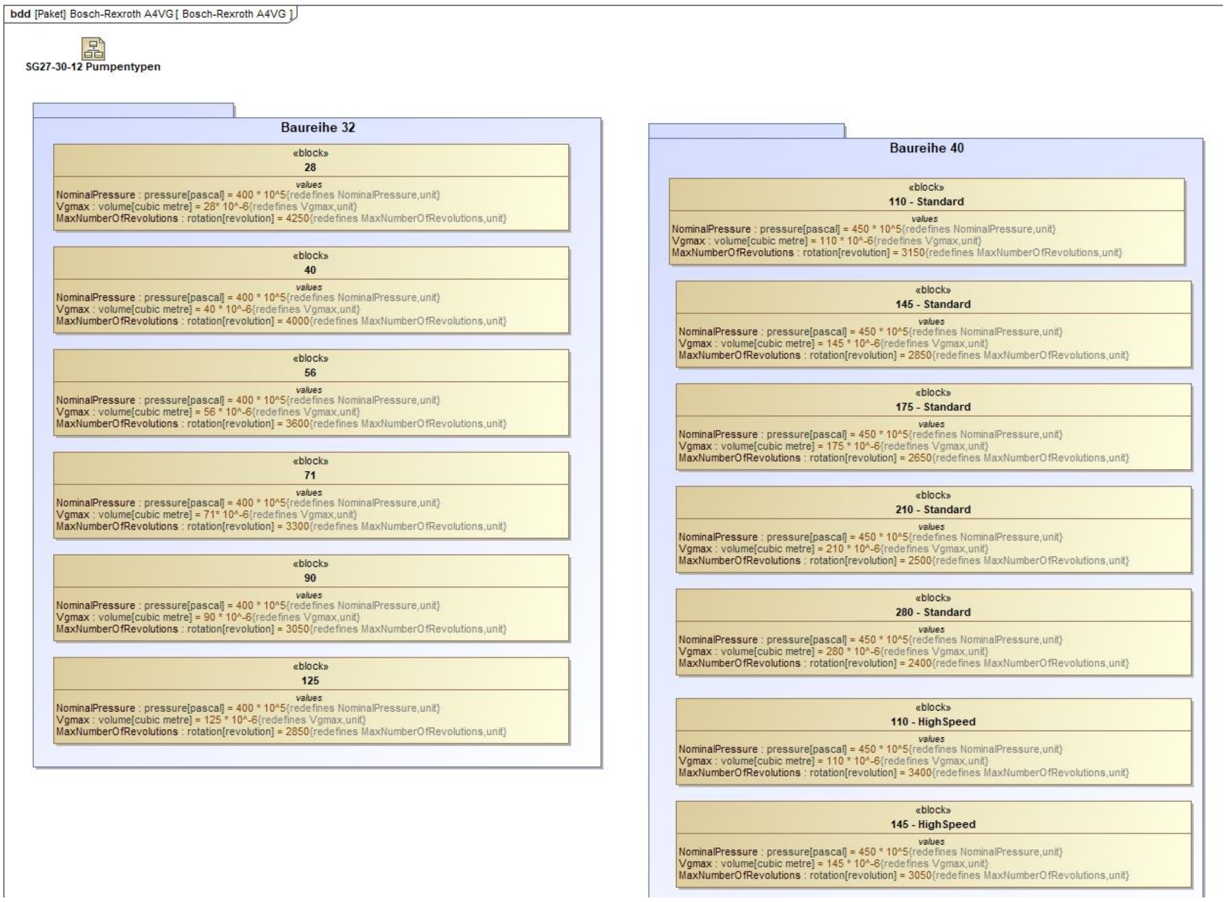
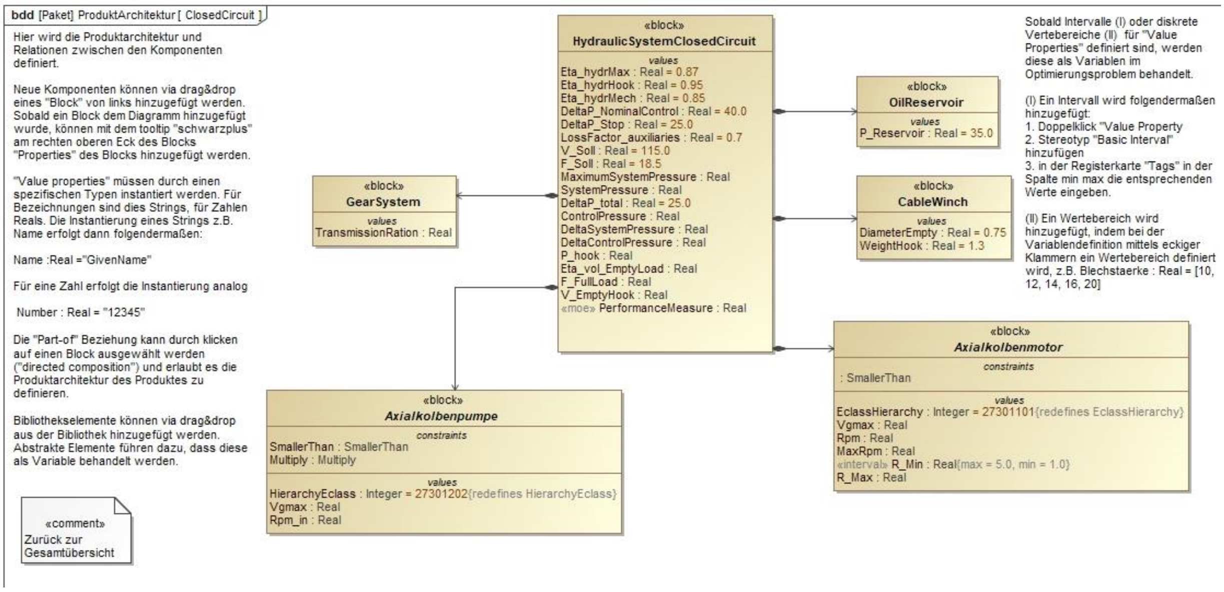


Figure 0-26: BDD defining specific types of hydraulic pumps, Scenario 1



Hier wird die Produktarchitektur und Relationen zwischen den Komponenten definiert.

Neue Komponenten können via drag&drop eines "Block" von links hinzugefügt werden. Sobald ein Block dem Diagramm hinzugefügt wurde, können mit dem tooltip "schwarzplus" am rechten oberen Eck des Blocks "Properties" des Blocks hinzugefügt werden.

"Value properties" müssen durch einen spezifischen Typen instantiert werden. Für Bezeichnungen sind dies Strings, für Zahlen Reals. Die Instantierung eines Strings z.B. Name erfolgt dann folgendermaßen:

Name :Real ="GivenName"

Für eine Zahl erfolgt die Instantierung analog

Number : Real = "12345"

Die "Part-of" Beziehung kann durch klicken auf einen Block ausgewählt werden ("directed composition") und erlaubt es die Produktarchitektur des Produktes zu definieren.

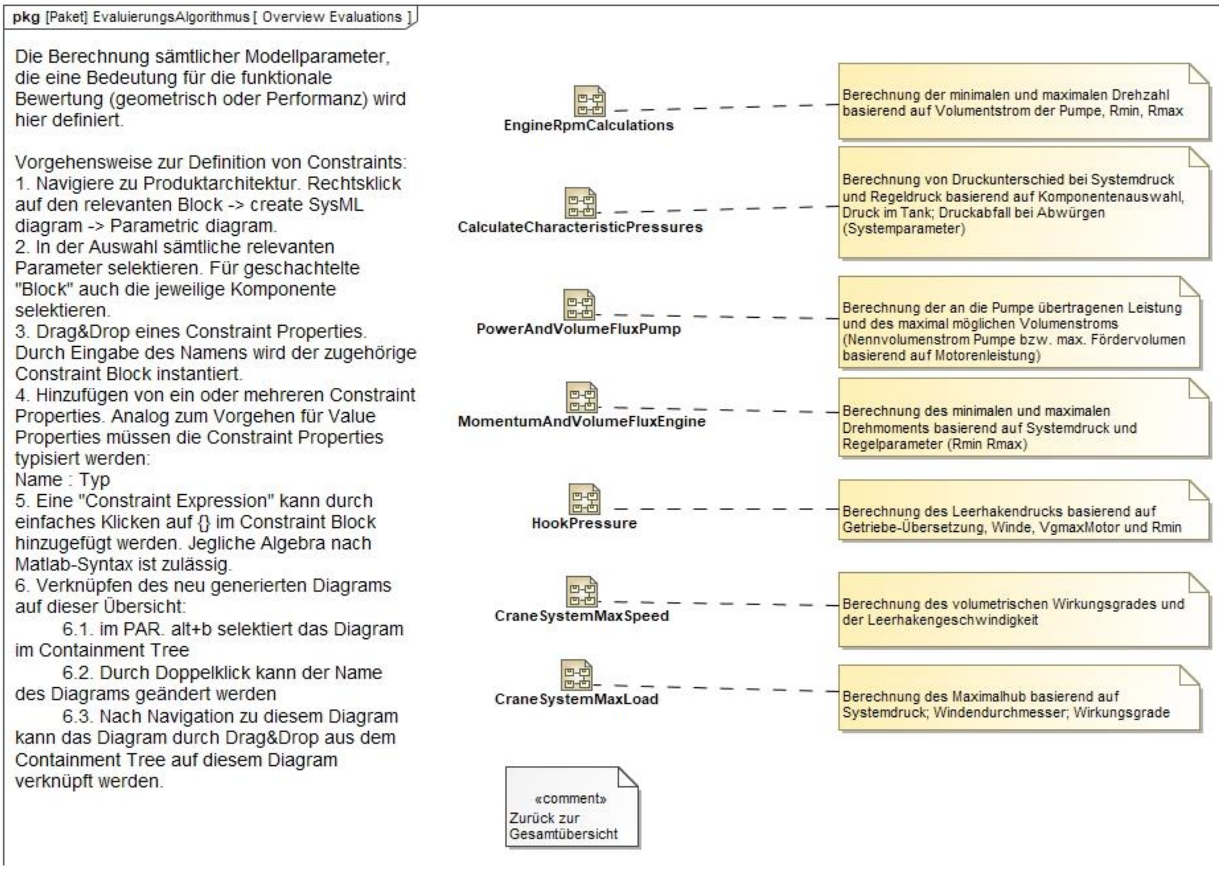
Bibliothekselemente können via drag&drop aus der Bibliothek hinzugefügt werden. Abstrakte Elemente führen dazu, dass diese als Variable behandelt werden.

Sobald Intervalle (I) oder diskrete Vertebereiche (II) für "Value Properties" definiert sind, werden diese als Variablen im Optimierungsproblem behandelt.

(I) Ein Intervall wird folgendermaßen hinzugefügt:
1. Doppelklick "Value Property
2. Stereotyp "Basic Interval" hinzufügen
3. in der Registerkarte "Tags" in der Spalte min max die entsprechenden Werte eingeben.

(II) Ein Wertebereich wird hinzugefügt, indem bei der Variablendefinition mittels eckiger Klammern ein Wertebereich definiert wird, z.B. Blechstaerke : Real = [10, 12, 14, 16, 20]

Figure 0-27: BDD representing product architecture of the closed circuit indicating the variables based on abstract blocks, Scenario 1



Die Berechnung sämtlicher Modellparameter, die eine Bedeutung für die funktionale Bewertung (geometrisch oder Performanz) wird hier definiert.

Vorgehensweise zur Definition von Constraints:
1. Navigiere zu Produktarchitektur. Rechtsklick auf den relevanten Block -> create SysML diagram -> Parametric diagram.
2. In der Auswahl sämtliche relevanten Parameter selektieren. Für geschachtelte "Block" auch die jeweilige Komponente selektieren.
3. Drag&Drop eines Constraint Properties. Durch Eingabe des Namens wird der zugehörige Constraint Block instantiert.
4. Hinzufügen von ein oder mehreren Constraint Properties. Analog zum Vorgehen für Value Properties müssen die Constraint Properties typisiert werden:
Name : Typ
5. Eine "Constraint Expression" kann durch einfaches Klicken auf {} im Constraint Block hinzugefügt werden. Jegliche Algebra nach Matlab-Syntax ist zulässig.
6. Verknüpfen des neu generierten Diagrams auf dieser Übersicht:
    6.1. im PAR. alt+b selektiert das Diagram im Containment Tree
    6.2. Durch Doppelklick kann der Name des Diagrams geändert werden
    6.3. Nach Navigation zu diesem Diagram kann das Diagram durch Drag&Drop aus dem Containment Tree auf diesem Diagram verknüpft werden.

Berechnung der minimalen und maximalen Drehzahl basierend auf Volumentstrom der Pumpe, Rmin, Rmax

Berechnung von Druckunterschied bei Systemdruck und Regeldruck basierend auf Komponentenauswahl, Druck im Tank; Druckabfall bei Abwürgen (Systemparameter)

Berechnung der an die Pumpe übertragenen Leistung und des maximal möglichen Volumenstroms (Nennvolumenstrom Pumpe bzw. max. Fördervolumen basierend auf Motorenleistung)

Berechnung des minimalen und maximalen Drehmoments basierend auf Systemdruck und Regelparameter (Rmin Rmax)

Berechnung des Leerhakendrucks basierend auf Getriebe-Übersetzung, Winde, VgmaxMotor und Rmin

Berechnung des volumetrischen Wirkungsgrades und der Leerhakengeschwindigkeit

Berechnung des Maximalhub basierend auf Systemdruck; Windendurchmesser; Wirkungsgrade

Figure 0-28: PKG enabling navigation among different types of constraints required for evaluation of performance for Scenario 1

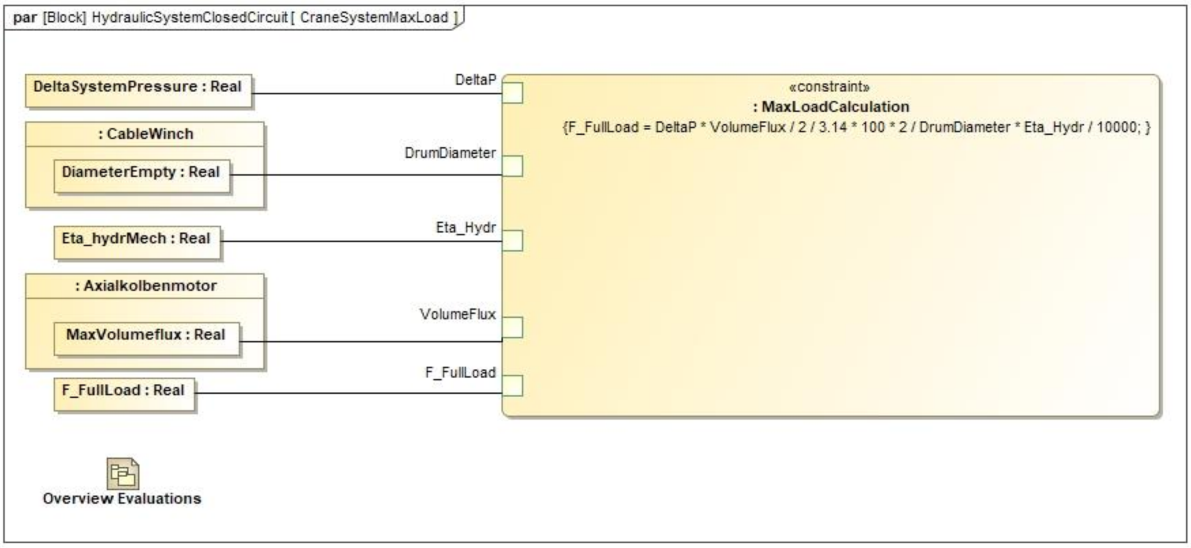


Figure 0-29: PAR indicating the parametric relation among parameters for calculation of the maximum load. This constraint is displayed representatively for the constraints indicated in Figure 0-28, Scenario 1

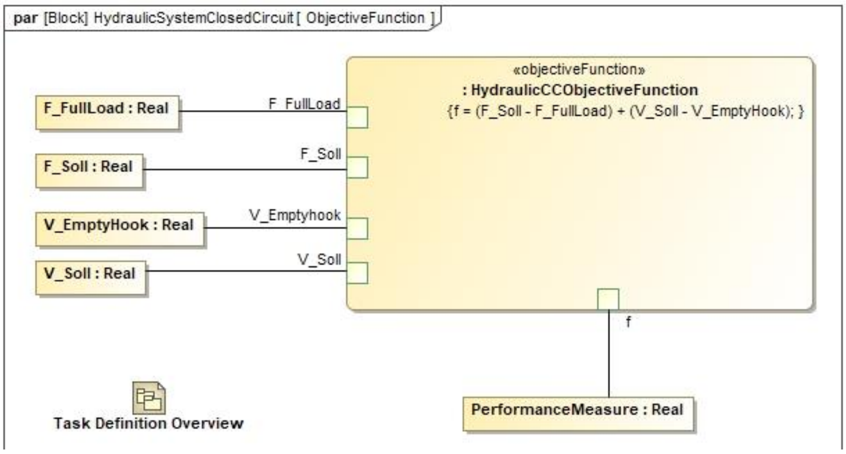


Figure 0-30: PAR indicating the objective function prior to the required modifications for Scenario 2

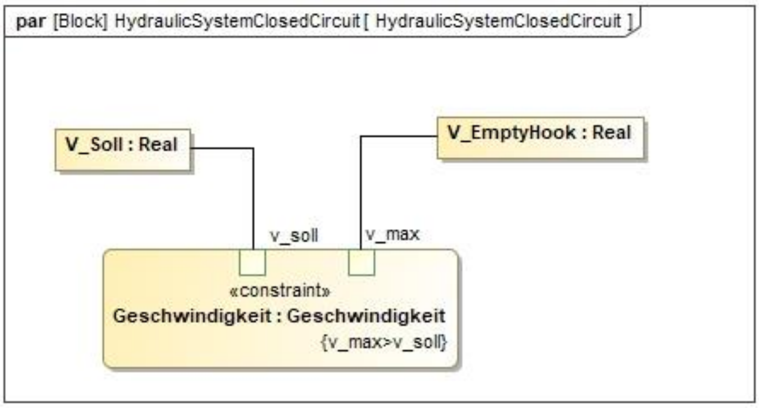


Figure 0-31: PAR indicating constraint of velocity required for accomplishing the tasks posed for Scenario 1

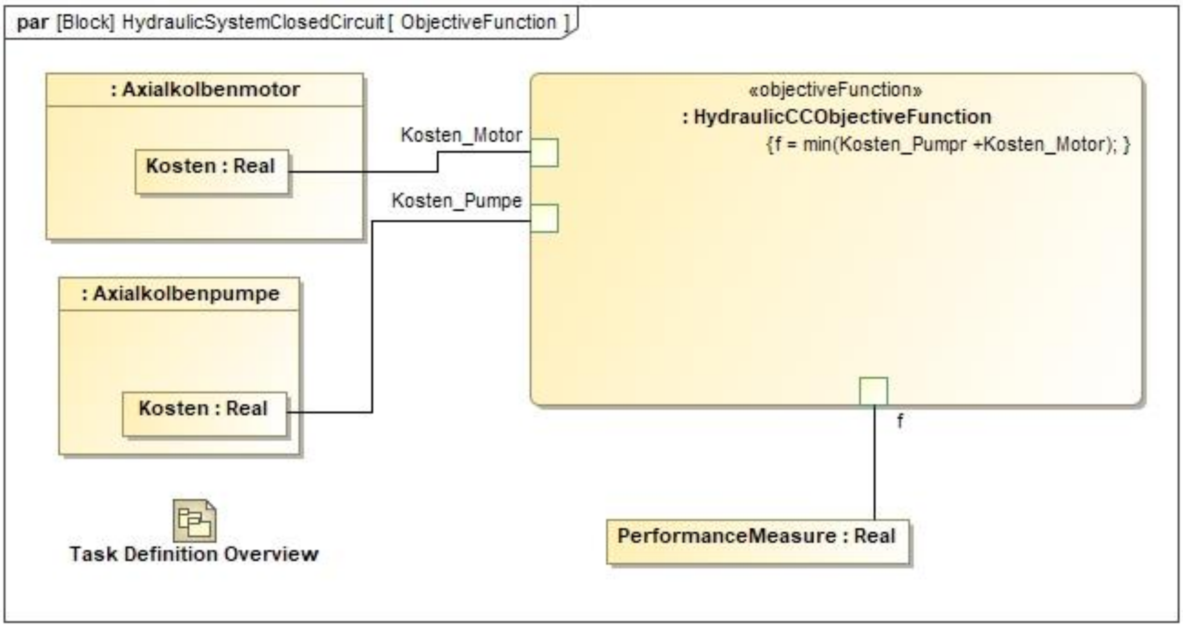
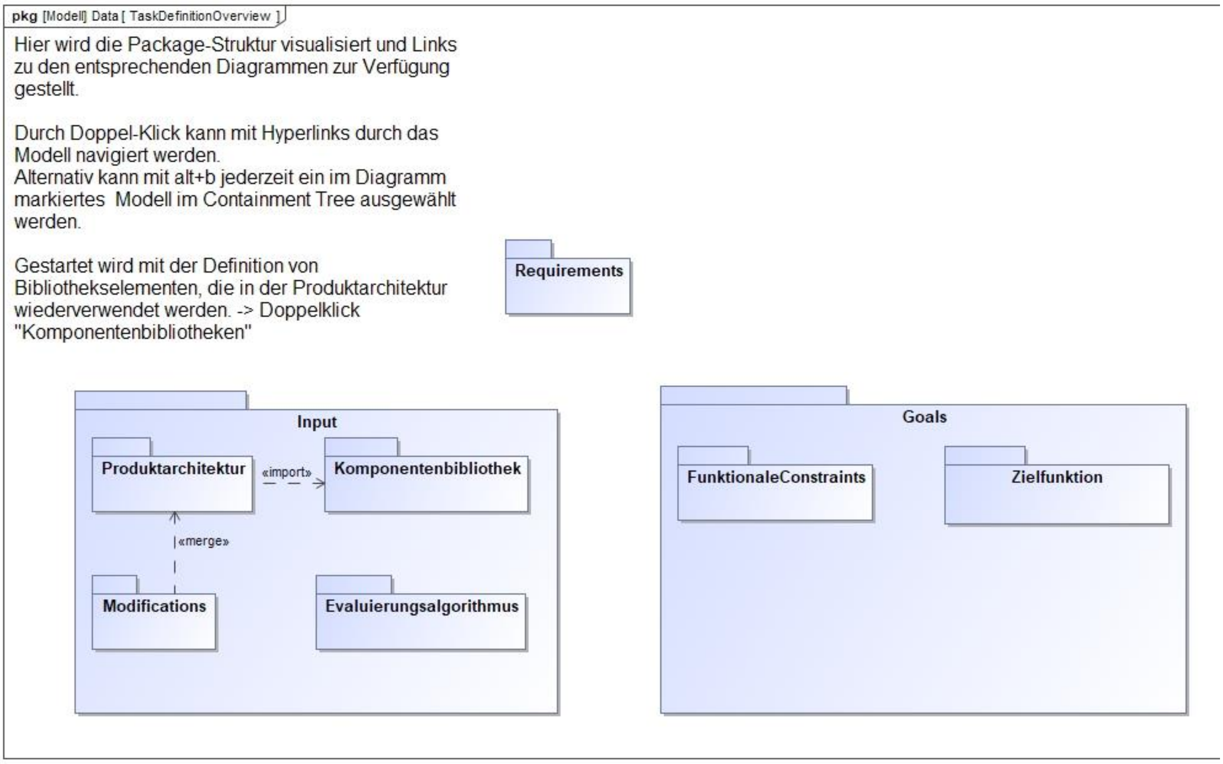


Figure 0-32: PAR indicating updated objective function for Scenario 1 after adding additional properties for costs of components.

Scenario 2 – Hydraulic Backpack



Hier wird die Package-Struktur visualisiert und Links zu den entsprechenden Diagrammen zur Verfügung gestellt.

Durch Doppel-Klick kann mit Hyperlinks durch das Modell navigiert werden.
Alternativ kann mit alt+b jederzeit ein im Diagramm markiertes Modell im Containment Tree ausgewählt werden.

Gestartet wird mit der Definition von Bibliothekselementen, die in der Produktarchitektur wiederverwendet werden. -> Doppelklick "Komponentenbibliotheken"

Figure 0-33: PKG for navigation within design automation task formalization of Scenario 2

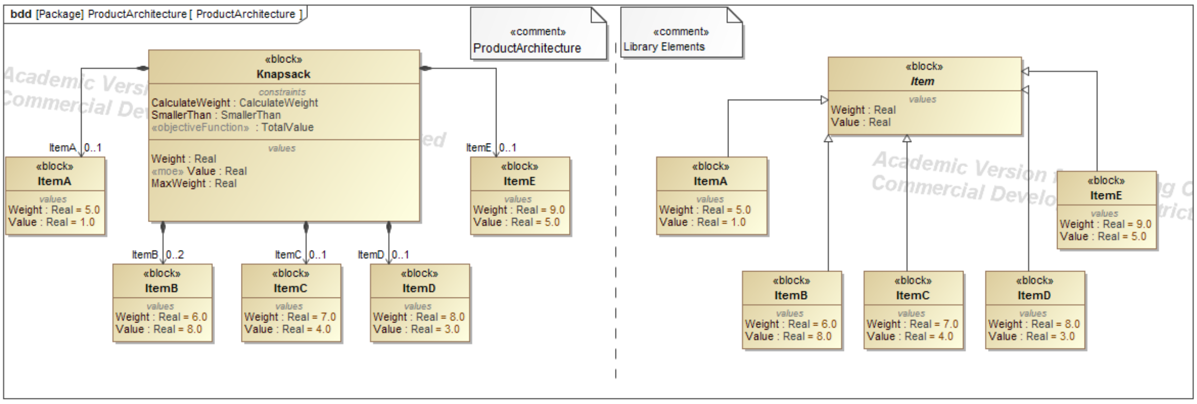


Figure 0-34: BDD for definition of component library (right) and product architecture (left) for Scenario 2

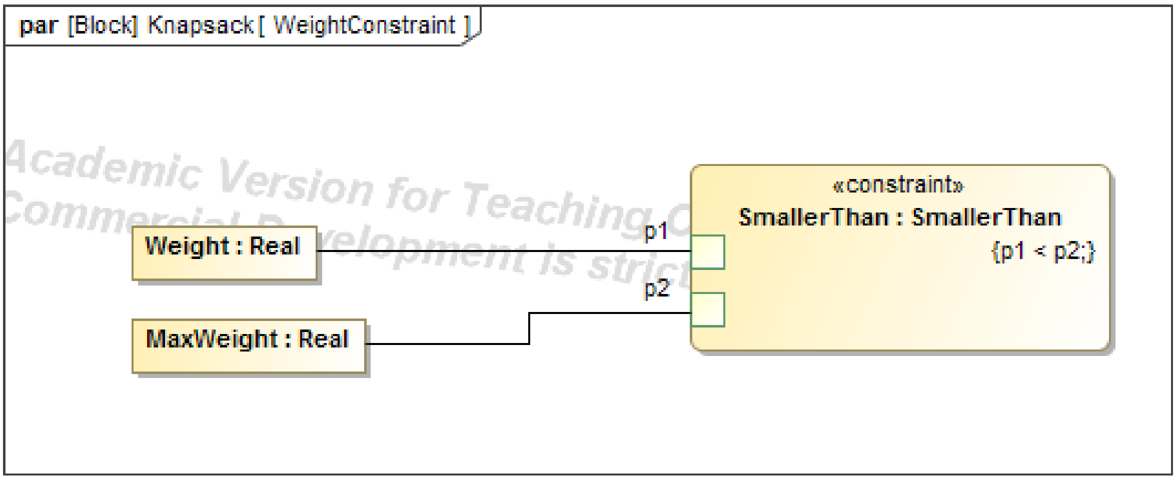


Figure 0-35: PAR representing weight constraint for the backpack, Scenario 2

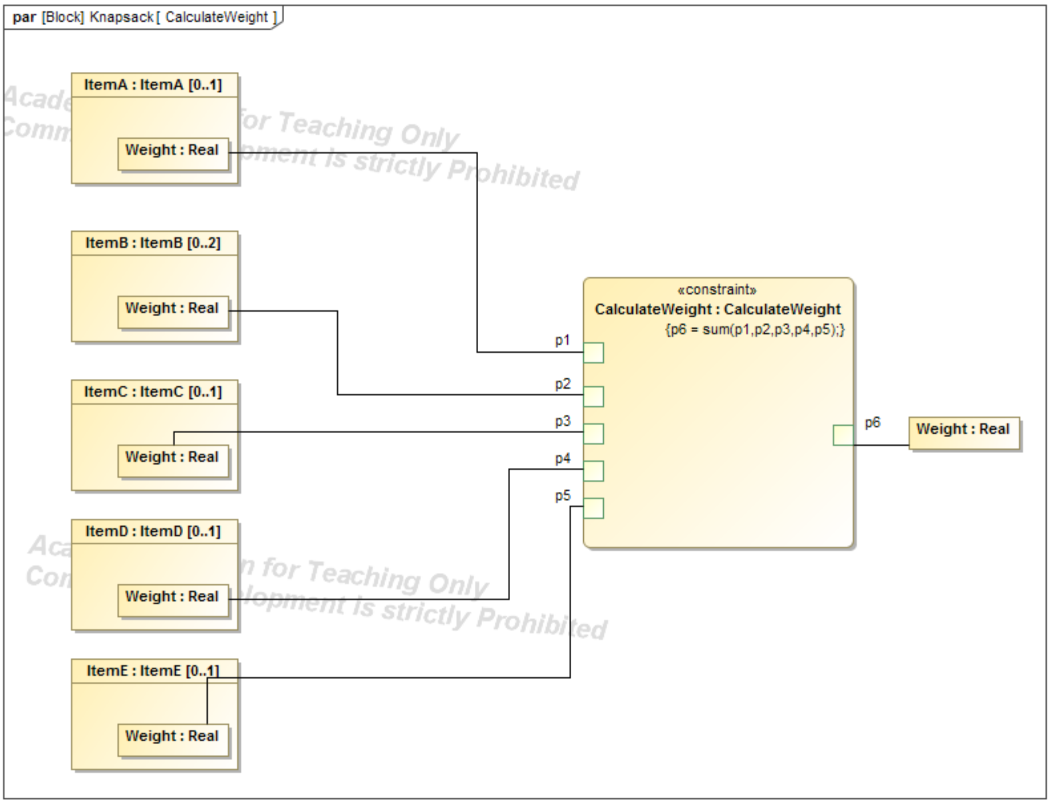


Figure 0-36: PAR for calculation of total weight of backpack, Scenario 2

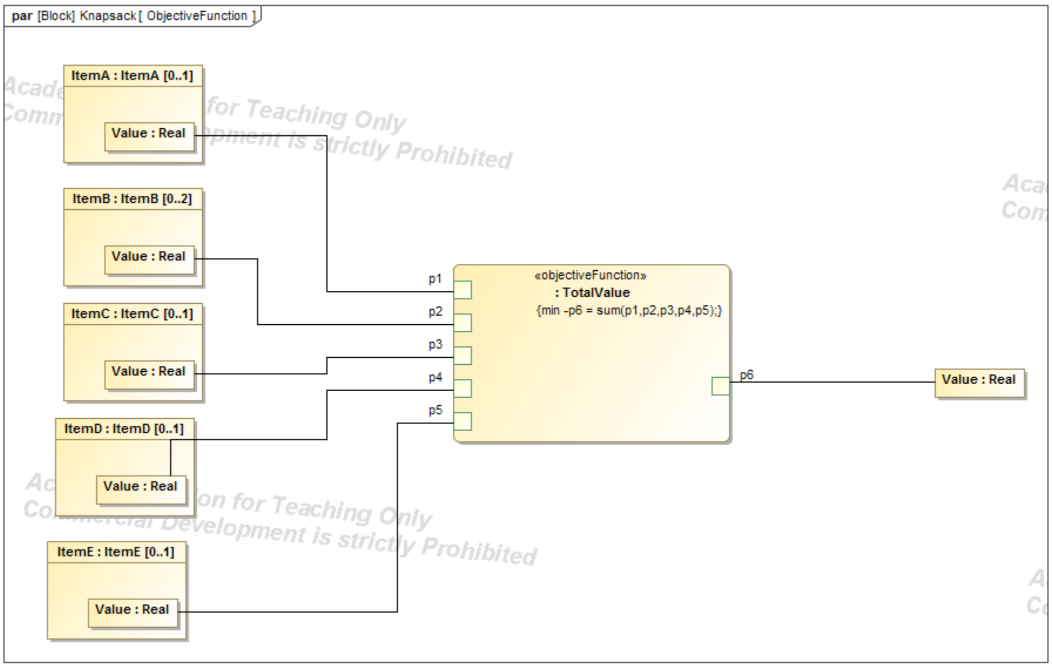


Figure 0-37: PAR defining the objective function for calculation of the total value of items stored in the backpack, Scenario 2