

TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Produktentwicklung

Structure-based System Dynamics Analysis of Engineering Design Processes

Daniel Kasperek

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität
München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing Manfred Hajek
Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. Udo Lindemann
2. Prof. Dr. Tyson R. Browning, Texas Christian
University, Fort Worth, Texas, USA

Die Dissertation wurde am 30.03.2016 bei der Technischen Universität München
eingereicht und durch die Fakultät für Maschinenwesen
am 08.07.2016 angenommen.

Acknowledgements

This work results from my occupation as a scientific researcher at the Institute of Product Development at the Technical University of Munich from May 2012 until March 2016.

Special thanks go to my supervisor Prof. Dr.-Ing. Udo Lindemann for the intense support of my research and his confidence in my ideas. Especially, his provision of scientific freedom and supporting reviews provided the basis for the successful compilation of the present work.

I would like to extend deepest gratitude to my second examiner Prof. Tyson Browning for the continuous guidance, inspiration and encouragement. The interaction was a sheer pleasure from the first email encounter until the finalization of the thesis and I feel very fortunate for having had so many conversations and discussions. Also, I want to thank Prof. Dr.-Ing. Manfred Hajek for accepting the chair of the examination board and for the organizational handling of the dissertation process.

I want to thank all my colleagues at the Institute of Product Development, many of whom have become close friends through our collaboration in projects, research and teaching. When I think back I see many moments of inspiration, joy and commitment. I would especially like to thank the SFB / CRC 768 subproject A2 team Sebastian Maisenbacher, Andreas Kohn and Maik Maurer. The discussions and work together provided closer insights into the dimensions of complexity and largely contributed to my research results.

In particular the support of my friends from the Kieferngarten-Stammtisch and especially Michael Roth needs to be mentioned. I have gained many valuable ideas and much energy during our ongoing discussions and our fruitful collaboration in different contexts. Finally, I want to thank Christopher Münzberg, Landon Raitz and again Michael Roth who helped me proofread this work and finalize all the little details.

I want to thank all my student assistants as well as those who worked out their thesis under my supervision; the obtained insights and results mainly contributed to my research work. Christian Lichtenberg, Markus Lindinger, Lukas Bermond, Elisabeth Schmidt, Sandra Berger, Alexander Bierig and Daniel Schenk deserve special mentioning.

Special thanks go to my wife, who always encouraged me in my work and offered patience and comprehension in stressful times. Also, I want to mention my daughter Leonie, who helped me to recognize the important things in life. Finally, I am very grateful to my parents for the extraordinary and permanent support. They offered me the chance for obtaining unforgettable experiences that incorporated important decisions in my life.

Garching, 08.07.2016

Daniel Kasperek

This dissertation contains small-sized illustrations. For the magnification of details, the author refers to the digitally published version of this dissertation on mediaTUM, the TUM's media server.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of Technical University of Munich's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

*“Remember that all models are wrong;
the practical question is how wrong do they have to be to not be useful.”
Box & Draper, 1987*

THE FOLLOWING PUBLICATIONS ARE PART OF THE WORK PRESENTED IN THIS THESIS:

- Kasperek, D., Berger, S., Maisenbacher, S., Lindemann, U., & Maurer, M. S. (2015). Structure-based System Dynamics Analysis - A Case Study of Benchmarking Process Optimization. In *International Conference on Engineering Design, ICED15* (pp. 1–10). Milano, Italy.
- Kasperek, D., Bermond, L., Maisenbacher, S., Zaggl, M. A., Raasch, C., & Maurer, M. S. (2015). Structure-based System Dynamics Analysis – A Case Study of Line Process Optimization. In *9th Annual IEEE International Systems Conference, IEEE SysCon15*. Vancouver, Canada.
- Kasperek, D., Chucholowski, N., Maisenbacher, S., Lindemann, U., & Maurer, M. S. (2014). A Method for Impact Analysis of Cyclic Changes within Innovation Processes of PSS. In *6th CIRP Conference on Industrial Product-Service Systems* (pp. 205–210). Windsor, Canada.
- Kasperek, D., Fink, S., Maisenbacher, S., Bauer, W., & Maurer, M. S. (2014). Assessing the informative value of complexity metrics within design structure matrices in early development phases of complex systems. In *8th Annual IEEE International Systems Conference, IEEE SysCon14* (pp. 351–356). Ottawa, Canada.
- Kasperek, D., Kohn, A., & Maurer, M. S. (2013). Identifying Uncertainties within Structural Complexity Management. In *International Conference on Engineering Design, ICED13*. (pp. 1–10). Seoul, South Korea: Design Society.
- Kasperek, D., Lichtenberg, C., Maisenbacher, S., Hollauer, C., Omer, M., & Maurer, M. S. (2015). Structure-based System Dynamics Analysis of Engineering Design Processes - A case study. In *9th Annual IEEE International Systems Conference, IEEE SysCon15*. Vancouver, Canada.
- Kasperek, D., Lindinger, M., Maisenbacher, S., & Maurer, M. S. (2014). A structure - based System Dynamics Approach for Assessing Engineering Design Processes. In *International System Dynamics Conference*. Delft, The Netherlands.
- Kasperek, D., Maisenbacher, S., Kohn, A., Lindemann, U., & Maurer, M. S. (2015). Increasing the reproducibility of structural modelling. *Journal of Engineering Design*, 26(7-9), 259–281. <http://doi.org/10.1080/09544828.2015.1026883>
- Kasperek, D., Maisenbacher, S., & Maurer, M. S. (2014a). Assessing Process Sequences in Engineering Design by DSM-based System Dynamics Simulation. In *4th International Engineering Systems Symposium, CESUN 2014*. Hoboken, N.J.
- Kasperek, D., Maisenbacher, S., & Maurer, M. S. (2014b). Structure-based analysis of dynamic engineering process behavior. In *8th Annual IEEE International Systems Conference, SysCon 2014 - Proceedings* (pp. 83–88). Ottawa, Canada.
- Kasperek, D., Maisenbacher, S., & Maurer, M. S. (2015). Structure-based Compilation of System Dynamics Models for Assessing Engineering Design Process Behavior. *The Journal of Modern Project Management*, 3(2), 101–107.
- Kasperek, D., & Maurer, M. S. (2013). Coupling Structural Complexity Management and System Dynamics to represent the dynamic behavior of product development processes. In *7th Annual IEEE International Systems Conference, IEEE SysCon13* (pp. 414–419). Orlando, USA: IEEE.

Lichtenberg, C., Kasperek, D., Maisenbacher, S., & Maurer, M. S. (2013). Strukturbasierte Modellierung und Bewertung von Entwicklungsprozessen von Produkt-Service Systemen. In *Tag des Systems Engineering* (pp. 219–228). Stuttgart, Germany.

Schmidt, E., Kasperek, D., & Maurer, M. S. (2015). A Guideline for Adapted System Dynamics Modeling of Rework Cycles in Engineering Design Processes. In W. J. C. V. Richard Curran, Nel Wognum, Milton Borsato, Josip Stjepandić (Ed.), *Transdisciplinary Lifecycle Analysis of Systems* (Volume 2:, pp. 357 – 366). Delft, The Netherlands.

CONTENT

1. Introduction	5
1.1 Initial situation and problem description	5
1.2 Objective, aims, research question and contribution of the thesis	7
1.3 Research method and environment	9
1.4 Thesis structure	11
2. Engineering design processes	13
2.1 Characteristics of engineering design processes	13
2.2 Engineering design processes as complex systems	18
2.2.1 Terms and definitions	18
2.2.2 Structural complexity	21
2.2.3 Dynamic complexity	22
2.2.4 The relationship of complexity, uncertainty and human behavior	23
2.3 Modeling of engineering design processes	26
2.3.1 Classes of engineering design process models	27
2.3.2 Modeling of structural complexity	29
2.3.3 Modeling of dynamic complexity	34
2.4 Early phase of engineering design processes	36
2.5 Existing models combining structural and dynamic complexity of engineering design processes	37
3. Case studies	43
3.1 Structure of the case studies and overview	43
3.1.1 Structure of the case studies	43
3.1.2 Overview of the case studies	45
3.2 Case study 1: Engineering design process of the PSSycle	45
3.2.1 Description of the case	45
3.2.2 Stage 1: Research question	45
3.2.3 Stage 2: Instrument development	46

3.2.4	Stage 3: Data gathering	46
3.2.5	Stage 4: Analyze Data	48
3.2.6	Stage 5: Disseminate	55
3.2.7	Implications of the case study	57
3.3	Case study 2: Line process optimization at MAN Bus & Coach	58
3.3.1	Description of the case	58
3.3.2	Stage 1: Research question	59
3.3.3	Stage 2: Instrument development	59
3.3.4	Stage 3: Data gathering	59
3.3.5	Stage 4: Analyze Data	61
3.3.6	Stage 5: Disseminate	67
3.3.7	Implications of the case study	69
3.4	Case study 3: Benchmarking process analysis at MAN Truck & Bus AG	71
3.4.1	Description of the case	71
3.4.2	Stage 1: Research question	72
3.4.3	Stage 2: Instrument development	72
3.4.4	Stage 3: Data gathering	73
3.4.5	Stage 4: Analyze Data	76
3.4.6	Stage 5: Disseminate	81
3.4.7	Implications of the Case Study	82
3.5	Case study 4: CALA construction process	83
3.5.1	Description of the case	83
3.5.2	Stage 1: Research question	84
3.5.3	Stage 2: Instrument development	84
3.5.4	Stage 3: Data gathering	84
3.5.5	Stage 4: Analyze Data	89
3.5.6	Stage 5: Disseminate	102
3.5.7	Implications of the Case Study	103
3.6	Summary of the case studies	104
4.	Framework for structure-based System Dynamics Analysis of Engineering Design Processes	106
4.1	Summary of the framework	106

4.2	Step 1: MDM modeling	109
4.3	Step 2: Transformation and quantification	112
4.3.1	Step 2a: Qualitative System Dynamics modeling	113
4.3.2	Step 2b: Quantitative System Dynamics modeling	127
4.4	Step 3: Testing of the System Dynamics model	136
4.5	Step 4: Simulative analysis	140
4.6	Step 5: Design application	145
5.	Evaluation of the framework	149
5.1	Initial situation and description of the use case	149
5.2	Objectives and instrumental development	150
5.3	Step1: MDM modeling	151
5.4	Step 2a: Qualitative System Dynamics modeling	153
5.5	Step 2b: Quantitative System Dynamics modeling	153
5.6	Step 3: Testing of the model	156
5.7	Step 4: Simulative analysis	157
5.8	Step 5: Design application	159
5.9	Evaluation of results and feedback from the industry partner	162
6.	General points of discussion	164
7.	Conclusion and outlook	166
7.1	Conclusion	166
7.2	Outlook	168
8.	References	171
9.	Appendix	189
9.1	Identified literature of literature search	189
9.2	Selection of relevant challenges from Sterman (2000)	194
9.3	Supervised student theses	195
10.	List of dissertations	196

1. Introduction

From a systems theory perspective engineering design processes are structurally and dynamically complex systems. In many engineering design situations it is necessary to understand both the structural and dynamic complexity of the process as well as their interrelations. This thesis aspires to enable a combined consideration of the domains of structural and dynamic complexity for engineering design processes. The introductory chapter illustrates the initial situation and its challenges, and it outlines the objectives of the thesis at hand. Furthermore, the research environment and relevant aspects of research methodology are elaborated and the structure of the thesis is presented.

1.1 Initial situation and problem description

The aim of engineering design process management can be described as a better definition and control of this processes with respect to the “three sacred cows”: time, quality and budget (Kneuper, 2003; Kreimeyer, 2009; Project Management Institute Inc, 2013).

The common measures of engineering design processes for the “sacred cows” are time-to-market, product performance and development cost. However, these performance measures may conflict with each other: the pressure to shorten time-to-market can conflict with maximizing product performance (Griffin & Page, 1993; Lilien & Yoon, 1990) and induce trade-offs between development cost and the objective to meet product performance (Bajaj et al., 2004; G. R. Smith et al., 1999). (Le, 2013)

To achieve a balanced outcome of all three performance measures, the interrelationships between them require effective process management. Effective management, however, requires thorough understanding of process behavior and potential influences on performances. (Le, 2013)

In contrast, engineering design processes are not repeatable, and they are inherently unpredictable in so far as completing an activity may result in a less complete state (Wynn et al., 2003).

This thesis, in accordance with Blessing (1994), represents the early phase of engineering design processes as the phases of planning and task clarification within the engineering design process model of Pahl (2007, p. 130). Especially during the early phase of engineering design processes uncertainty is often prevailing (Lévárdy & Browning, 2009; Wiebel et al., 2013). Engineering design processes can therefore also be seen as complex systems (e.g., Smith & Morrow, 1999).

Furthermore, the term uncertainty is used according to Lévárdy & Browning (2009) who state that uncertainty can be characterized as a lack of knowledge about a problem at the time of making a decision affecting its solution. A distinction can be made between foreseen and unforeseen uncertainty (also called ambiguity). Foreseen uncertainty can be categorized as knowing that you do not know and unforeseen uncertainty as not knowing that you do not know. (Lévárdy & Browning, 2009; Sarbacker & Ishii, 1997; Schrader et al., 1993)

As ambiguity characterizes engineering design processes, the traditional methods and tools of project management provide less value. Therefore, the attitudes and aptitudes of project managers and participants must change (De Neufville, 2004; Loch et al., 2006; Wiebel et al., 2013) along with their models, methods and tools. (Lévárdy & Browning, 2009)

While unforeseen uncertainty is a source of risk for engineering design processes, it can also provide opportunities for organizations capable of effectively sensing the endogenous and exogenous changes and adapting to the changed conditions (Haeckel, 1999). Engineering design processes which are capable of coevolving with their environments and dynamic stakeholder needs can profit from the accelerating pace of change in market needs (Dougherty, 2001; Eisenhardt & Tabrizi, 1995). (Lévárdy & Browning, 2009)

This is in accordance with Rouse's (2007) stance that fundamental complex systems research should focus on approaches, technologies and tools to enable decision support for those who invest in, develop, operate and use complex systems. For the planners and managers of industrial engineering design processes, it is worthwhile to learn more about the dynamic process behavior in order to distribute resources appropriately and to calculate cost and scheduling (Kreimeyer & Lindemann, 2011). This does not only address complexity, but also complicatedness (or cognitive complexity) which is the more subjective, observer-dependent aspect of complexity as a system, regardless of its complexity, may appear more or less complicated depending on one's point of view (Browning & Ramasesh, 2015; Ramasesh & Browning, 2014).

Currently, various approaches are being applied to solving the complex issue of planning and managing engineering design processes. Oehmen et al. (2015) describe a concept of structural and dynamic complexity during the development and deployment of engineering systems, which is driven by the developing organization, the technology to be developed, uncertainty and human behavior. Geraldi et al. (2011, p. 985) especially point out the immediate interest in the nature of interaction between complexity dimensions and "...in particular, how a dynamic change in one of the dimensions will impact the others".

This thesis addresses this issue of the interactions between complexity dimensions of structural complexity and dynamic complexity. In particular, this thesis focusses on models of the engineering design process structure (also referred to as structural models or dependency models) in the form of Multiple-Domain Matrices (MDM), as introduced by Lindemann et al. (2009), and on models of the dynamics of an engineering design process in the form of System Dynamics models and simulations.

MDMs provide the opportunity to describe the dependencies of a system by identifying the relations of individual domains and elements. However, structural models show a fundamental limitation: as structural models usually describe systems at a certain point in time, only static dependencies can be mapped (Maurer, 2007; Oehmen et al., 2015). To examine the dynamic dependencies of a system and the effects of uncertainty, other approaches are more suitable. The approach chosen in a specific situation depends on the goal of the analysis: when it comes to analyzing an overall process setup, a more static analysis might help to find bottlenecks and other points that should be the focus of subsequent analysis (Kreimeyer & Lindemann, 2011), whereas a dynamic approach will help to improve current procedures and their efficiency during the operation process.

A common approach for analyzing the dynamics of engineering design processes is System Dynamics introduced by Forrester (1958). This approach provides a methodology to analyze and simulate the dynamic behavior of an abstracted system (Sterman, 2000). System Dynamics is used in this thesis due to its adaptable degree of abstraction, its management perspective and the preliminary work on engineering design processes within the System Dynamics community, such as Cooper (1980), Ford & Sterman (1998a) and Le (2013). The limitation of System Dynamics is the opposite of that for structural models, as System Dynamics does not provide the possibility to describe static aspects of the system (H. Meier & Boßlau, 2012). Additionally, Warren (2014) identifies the need for standard System Dynamics structures to offer an accessible und reliable source of guidance for developing System Dynamics models.

During the early phase of engineering design processes it is essential to consider the static structural complexity dimension and the dynamic complexity dimension at the same time. Currently, the problem is that the modeling approaches of these two dimensions – the static structural modeling approaches and approaches modeling the dynamics of engineering design processes – cannot be sufficiently combined to model the impact of changes of one dimension on other dimensions as demanded by Geraldi et al. (2011). Nevertheless, it is known from the literature that the relations between the different entities of an engineering design process, i.e. its structure, are the foundation of the engineering design process's behavior, i.e. its dynamics (Flurschein, 1977; Kreimeyer, 2009; Rechten, 1991; Wasson, 2006).

Previous results show that – in principle – dynamic behavior can be deduced from structural models (Biedermann et al., 2012). As an example, Biedermann et al. (2012) identified that structurally highly-connected components of an assembly cell have a very predictable behavior.

1.2 Objective, aims, research question and contribution of the thesis

A successful combination of the static-structural and the dynamic-behavioral views on the engineering design processes would allow for developing models of engineering design processes that lend themselves to experimentation. The objective of such multi-dimensional models of engineering design processes for experimentation is to support early phases of engineering design processes by:

- understanding the intrinsic complexity with its dimensions (i.e. structural and dynamic) and their interactions
- decreasing the perceived complicatedness
- uncovering knowable unknown unknowns.

This thesis aspires to enable a combined consideration of the dimensions of structural and dynamic complexity.

Additionally, uncertainty is considered in this thesis as the conditions of complexity, and uncertainty usually occur at the same time (Oehmen et al., 2015). In particular, this thesis focuses on known uncertainty related to the long timeframe through which planning assumptions have to be projected.

To enable a combined consideration of structural and dynamic complexity under uncertainty, this thesis aims to use structural models in the form of MDMs as a basis for constructing System Dynamics models to analyze the dynamics of engineering design processes under uncertainty. The MDMs shall be used to represent the structural complexity, while System Dynamics shall be used to represent the dynamic complexity and also to model the influence of uncertainty on engineering design processes. To combine both dimensions of complexity it is necessary to transform the structural information in System Dynamics to model the dynamic complexity, as well as to transpose the results of the dynamics analysis back into implications on the structure of the engineering design process. The early availability of the structure of the engineering design process in the overall process supports the early phases of planning and task clarification within the engineering design processes by prescriptive process analysis. An overview of the considered complexity dimensions and modeling approaches within this thesis is illustrated in Figure 1-1.

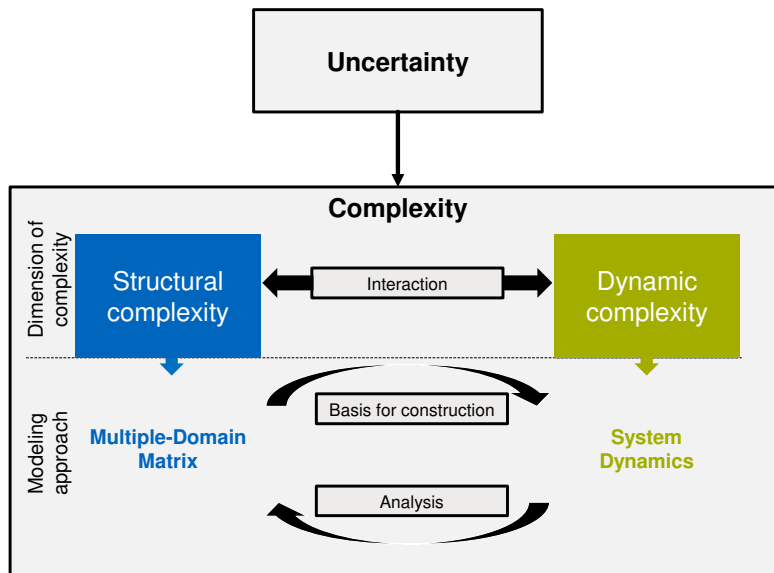


Figure 1-1 Considered complexity dimensions and modeling approaches within this thesis

Based on the specified aim, the overall research question can be formulated as follows:

How can structural models in form of MDMs be used as a basis to construct System Dynamics models for supporting the early phase of engineering design processes by prescriptive process analysis?

This question comprises several aspects that need to be answered for a thorough understanding and derivation of the further procedure. On the one hand the transformation from a MDM to a System Dynamics model needs to be elaborated (“Basis for construction” in Figure 1-1):

- Is the structural information from MDMs sufficient for System Dynamics modeling?
- How can MDMs be used as a basis for the construction of System Dynamics models?
- Which information can be incorporated in the MDMs?

Furthermore, it needs to be clarified how the results of the dynamics analysis in System Dynamics can be obtained and applied back on the engineering design process structure (“Analysis” in Figure 1-1):

- How can uncertainty be modelled within System Dynamics models?
- How can the structure of engineering design processes be analyzed with System Dynamics?

If the overall research question can be satisfactorily answered this would lead to the following contributions (Successful handling of “Interaction” in Figure 1-1):

- Enabling new analysis metrics for structural models based on insights from the dynamic behavioral view.
- Supporting the early identification of major risks of engineering design processes.
- Giving decision support for the design of engineering design processes through simulation experiments and building scenarios.
- Allowing the benchmarking of different structural designs of engineering design processes under consideration of dynamics and uncertainty.

1.3 Research method and environment

This subchapter provides an overview of the applied research methodology on which the work for this thesis is based. Further, the research environment which comprises the employment, project work, and experiences and resources of the author is presented.

The scientific course of action follows the design research methodology (DRM) according to Blessing & Chakrabarti (2009). DRM comprises a four-stage process as depicted in Figure 1-2, which can be performed sequentially or iteratively.

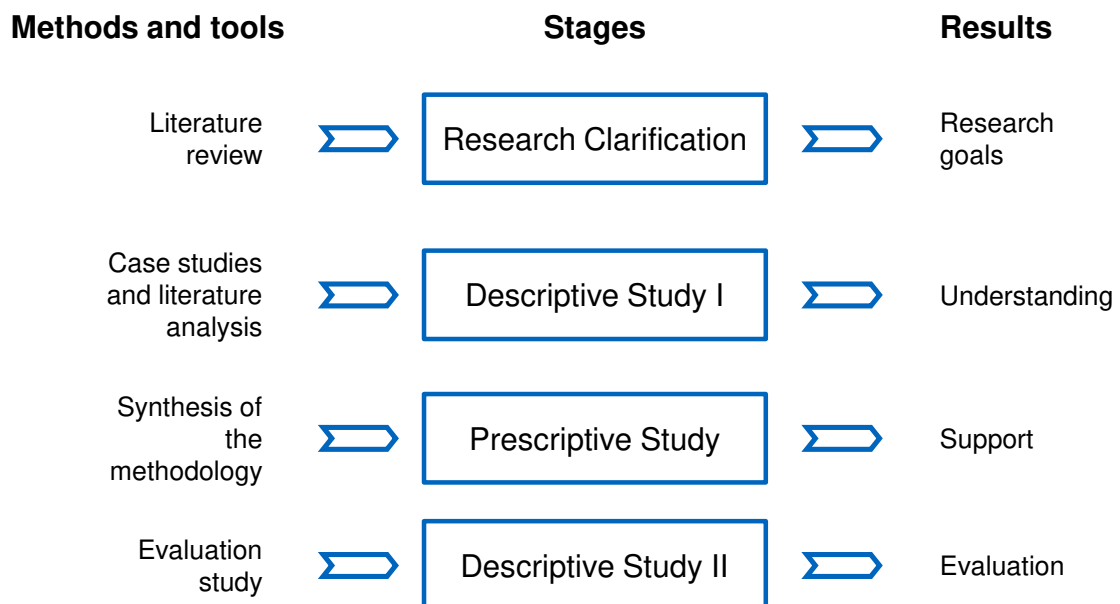


Figure 1-2: DRM research methodology (Blessing & Chakrabarti, 2009)

In the first stage of Research Clarification, the aim is to find some evidence or at least indications that support the assumptions in order to formulate realistic and worthwhile research objectives and aims. The main method is searching the literature for factors that influence task clarification and success of engineering design processes, in particular those factors that link

the two together. An initial description of the existing situation is developed based on these findings, as well as a description of the desired situation, in order to make the underlying assumptions for each of the descriptions explicit. The researchers formulate several criteria that could be used as measures against which the outcome of the research, (i.e., support for task clarification) could be evaluated. (Blessing & Chakrabarti, 2009) Two factors contributed substantially to the research clarification in the course of this thesis: The performed research on solely structure-based metrics for the analysis of the dynamics of engineering design processes revealed a lack of analysis possibilities. However, a literature review showed indications that the dynamics of engineering design processes can be deduced from the engineering design process structures.

In the Descriptive Study I, the literature is reviewed for more influencing factors to elaborate the initial description of the existing situation. The intention is to make the description detailed enough to determine which factor(s) should be addressed to improve task clarification as effectively and efficiently as possible. Complementary findings are discussed with experts in industry to obtain a better understanding of the existing situation before proceeding to the Prescriptive Study stage. (Blessing & Chakrabarti, 2009) The stage has been performed by conducting four industrial and academic case studies to understand the problem situation and on how to combine models of structural and dynamic complexity as well as a keyword-based literature analysis on existing approaches the course of this thesis.

In the Prescriptive Study stage, the increased understanding of the existing situation is used to develop support to overcome the problem situation examined in the Descriptive Study I. Based on the desired output and available inputs, methods and tools may be developed, modified, combined, or chosen. To this point it is not clear that the support has the desired effects, because of the many assumptions upon which the description of the desired situation and the development of the support have been based. (Blessing & Chakrabarti, 2009) The prescriptive study is represented by the developed structure-based System Dynamics Analysis framework, which allows analyzing the engineering design process structure for its correlation with the dynamics of the engineering design process and uncertainty.

The Descriptive Study II stage investigates the impact of the support and its ability to realize the desired situation. It is typically performed by empirical studies to gain an understanding of the actual use of the support. (Blessing & Chakrabarti, 2009) Blessing & Chakrabarti (2009) distinguish three forms of evaluation: support, applicability and success evaluation. Those are mainly performed through the application of the framework within the evaluation use case.

This thesis was generated during the author's employment as a research assistant at the Institute of Product Development at the Technical University of Munich. Major parts of the thesis are based upon work performed in the research project "A2 - Modeling and evaluating development relationships across disciplines" within the Collaborative Research Center CRC 768 / SFB 768 "Managing cycles in innovation processes – integrated development of product-service systems based on technical products" funded by the Deutsche Forschungsgemeinschaft (DFG).

The goal of CRC 768 / SFB 768 is the transdisciplinary development of models, methods, and tools for creating innovative product-service systems. The principal objectives are to improve the effectiveness and efficiency of innovation processes for product-service systems. (Technische Universität München, 2015)

The work was further supported by several student theses (as listed in appendix 9.3).

1.4 Thesis structure

The structure of the thesis is oriented on the four stages of the described DRM methodology (Blessing & Chakrabarti, 2009). The research clarification is presented in the form of the problem description and objective in chapter 1. Chapter 2 provides an overview about existing work on engineering design processes in relevant areas and comprises the first part of the Descriptive Study I, which especially outlines and elaborates on existing approaches in the field of structural and dynamic complexity of engineering design processes. Additionally, this chapter discusses the research gap. Based on the insights from literature in chapter 2, chapter 3 comprises the case study based second part of the Descriptive Study I. The developed support–representing the prescriptive study – is presented in chapter 4. Chapter 5 addresses the evaluation and therefore Descriptive Study II. An assessment of the support, applicability, and success of the developed framework is performed through the application of industrial use cases and expert interviews. Chapter 6 discusses the aspects of correctness, completeness, consistency and clearness of the framework as well as its relevance for academia and industry. The thesis closes with a conclusion and outlook, elaborating on what has been achieved and where further work is desirable. Figure 1-3 gives an overview of the structure of the thesis.

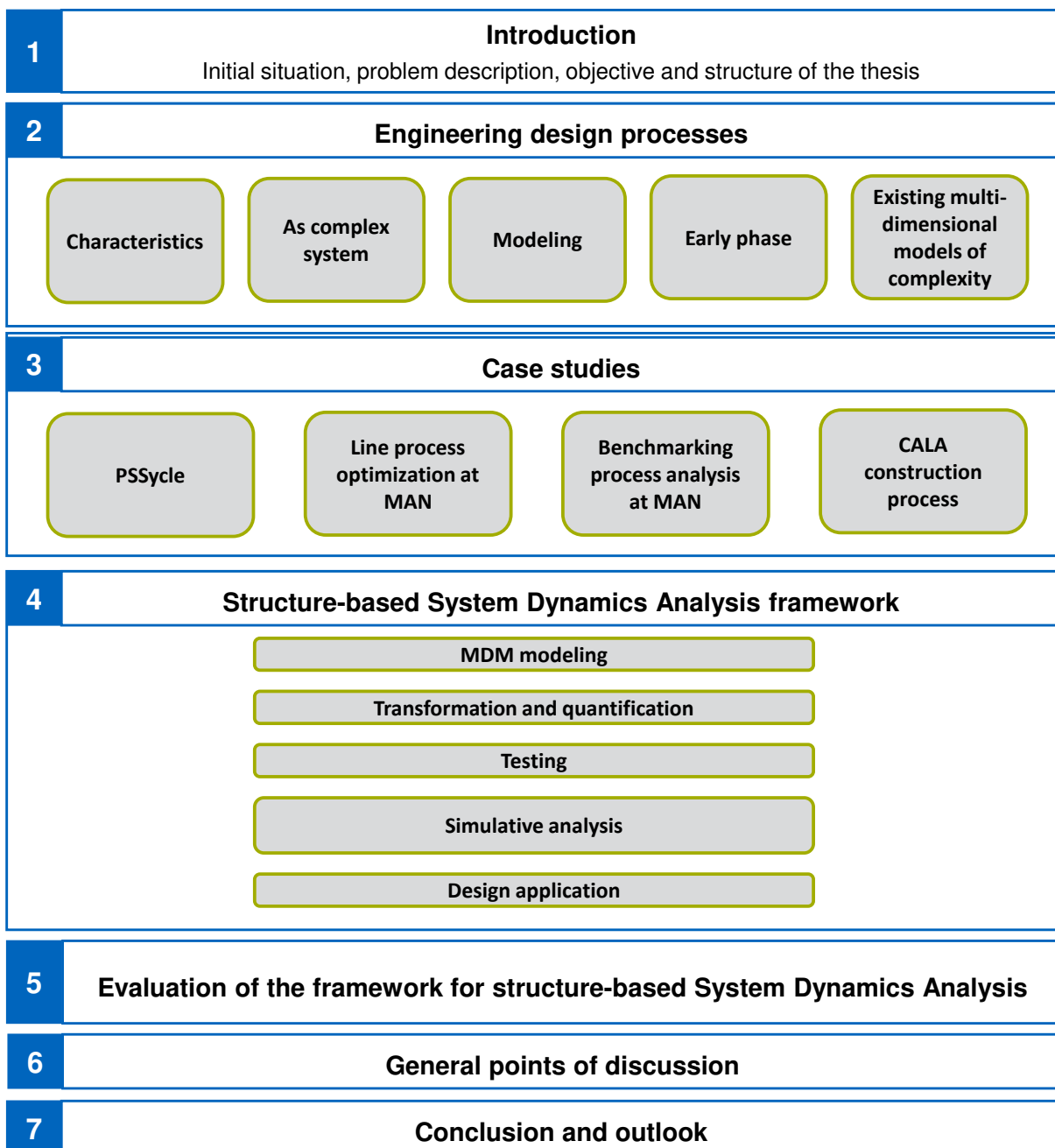


Figure 1-3 Structure of the thesis

2. Engineering design processes

Chapter 2 provides an overview about existing work on engineering design processes in relevant areas and comprises the first part of the Descriptive Study I. In particular it outlines and elaborates on existing approaches in the field of combining structural and dynamic complexity of engineering design processes, and it discusses the research gap.

2.1 Characteristics of engineering design processes

The development of products in engineering domains is an organizational capability and a competitive lever for companies (Maier & Störrle, 2011). One way to represent, understand, engineer, manage and improve engineering design is by taking a process-perspective, and in particular by using process modeling and creating process models (Kreimeyer, 2009; Maier & Störrle, 2011; Maurer, 2007).

Within this thesis, the process definition of Kreimeyer (2009, p. 63) is used which he developed based on Van Der Aalst & Van Hee (2004) and a comparison of other process definitions:

“A **process** consists of interdependent tasks that exchange information via artifacts. The process is enabled and supported by the purposeful allocation of resources and time-oriented constraints. All of these entities are interrelated, on the one hand, via the input-output relationships among tasks along the principal process flow, and, on the other hand, via other relationship types that generate the overall process network.”

Even though, there are several taxonomies on how to classify processes, one general notion is to distinguish between business processes and engineering design processes.

According to Becker et al. (2003, p. 4) Business Processes can be defined as stated in the following:

“A **business process** is a special process that is directed by the business objectives of a company and by the business environment. Essential features of a business process are interfaces to the business partners of the company (e.g., customers, suppliers). Examples of business processes are the order processing in a factory, the routing process of a retailer, or the credit assignment of a bank.”

For the term engineering design process the definition of Kreimeyer (2009, p. 64) is used:

“An **engineering design process** [...] is a process during which knowledge about an object is generated. As this object still necessitates designing, its nature is – at least in part – unknown. This generates uncertainty throughout the process that needs to be managed, and that causes an engineering design process to be much less deterministic than a business process.”

Vajna (2005) explicitly lists differences between business processes and engineering design processes (see Table 2-1). He identifies the need for process control for business processes and the need for process navigation for engineering design processes. Thereby navigation is meant in the way that the control and decision competence should always be left with the user.

Table 2-1 Differences between business processes and engineering design processes by Vajna (2005)

Business processes	Engineering design processes
Processes are fixed, rigid, have to be reproducible and checkable to 100%	Processes are dynamic, creative, chaotic; many loops and go-tos
Results have to be predictable	Results are not always predictable
Material, technologies, and tools are physical (e.g. in manufacturing) and/or completely described (e.g. in controlling)	Objects, concepts, ideas, designs, approaches, trials (and errors) are virtual and not always precise
Possibility of disruptions is low, because options and their respective environments are described precisely	Possibility of disruptions is high because of imperfect definitions and change requests
No need for dynamic reaction	There is definitive need for dynamic reaction capabilities

On a more detailed level Vajna (2005) identifies the following needs for an engineering design process navigation approach:

- Continuous monitoring of the process and prediction of possible bottlenecks.
- Creation and evaluation of potential process flow alternatives to overcome possible changes (for example, new requirements of a customer, a failed resource, a missed deadline, unforeseen disturbances) in real time.
- Offering of these alternatives to the user and allowing him to select the alternative he would prefer, then re-evaluating the process.

According to Vajna (2005), another metaphor for navigation is its description as a game of chess between the user in a company and the customer. In response to the possibility of customer changes and resource shortfalls during project execution, and in order to keep within time and budgetary constraints, the user has to update the project strategy continuously.

By conducting a literature study with a grounded-theory-inspired procedure, Maier & Störrle (2011) identifies nine characteristics of engineering design processes which they group into three categories and are listed in Table 2-2.

Table 2-2 Challenges and characteristics of engineering design processes identified by Maier & Störrle (2011)

Challenge	Engineering design process characteristic
Development	Engineering design processes are <ul style="list-style-type: none"> • complex • iterative, and • ill-defined.
Collaboration	Engineering design processes are embedded in an ecosystem of processes with multiple interdependencies and interactions between <ul style="list-style-type: none"> • different processes, • people involved, and • processes and organizations.
Products & Services	Engineering design processes are constrained and influenced by <ul style="list-style-type: none"> • the physical nature of the artefact developed, • economic and market constraints, and • legislative and regulatory constraints.

In the following the three categories and their nine characteristics will be explained.

Challenge: Development

Each engineering design process is concerned with the development of solutions for its unique set of underlying constraints and boundary conditions. Even though the development usually does not start from a blank sheet, the solutions to be developed are often new. (Maier & Störrle, 2011)

Complex: The engineering design processes of complex systems such as airplanes, jet-engines, automobiles are also highly complex (Ulrich & Eppinger, 2003). “The challenge for many designers is to maintain an adequate overview of a complex emerging product and its equally complex design process” (Eckert & Clarkson, 2005, p. 3). Oehmen et al. (2015) describe two dimensions of complexity that occur in the context of the engineering design processes: structural complexity, dynamic complexity, which both are driven by uncertainty and human behavior.

Geraldi et al. (2011) especially point out the current lack of understanding of interaction between complexity dimensions.

Iterative: All creative activities are iterative by nature. To sufficiently explore the design space and find a solution for a given set of constraints, various alternative paths and successive versions have to be pursued, elaborated, compared, split, fused, improved, evaluated, rejected, and reconsidered. (Maier & Störrle, 2011)

The structure of such a search process is described as iterative (e.g. Wynn et al. (2007))

Ill-defined: It is in the nature of engineering design processes that there are elements of uncertainty in their outcome (Grebici et al., 2008).

In fact, beginning from constraints and desires without precise goals, it is the very purpose of engineering design to develop and create solutions. There would be no need for creativity in the sense of a new or different solution if the result is completely defined at an operational level

of detail. Often, the problem or need is understood while the solution needs to be generated, and often there are rolling targets. (Maier & Störrle, 2011)

In this sense, engineering design problems and the processes to solve them are ill-defined (Rittel & Webber, 1984; Simon, 1962; Visser, 2006).

Challenge: Collaboration

Almost all instances of engineering design processes require the collaboration of many people. Most engineering design processes involve a great number of people which are distributed over space, time and organizations. Often, participants also exhibit strongly differing skill profiles. (Maier & Störrle, 2011)

Process-process interactions: “Most engineering design processes are intertwined with other engineering processes, embedded in other business processes in the organization and linked to a number of supplier companies’ process” (Eckert & Clarkson, 2005, p. 22). In fact, there are no processes which are isolated from triggering, supporting, or ensuing processes which are more or less well-defined and standardized in an organization. Engineering design processes are embedded in an ecosystem of other processes and exhibit multiple and complex interdependencies with positive and negative feedbacks. (Maier & Störrle, 2011)

Process-organization interactions: Processes are phenomena of the organizations which execute them. Besides official organizations and processes, there are always informal traditions, structures and external influences that influence engineering design processes. Organizations are also often collaborations of virtual development networks which additionally influences the processes. For example automotive supply chains are often global process-organizations interactions (Ulrich & Eppinger, 2003). (Maier & Störrle, 2011)

People-people interactions: “The design of a modern product, such as a car, requires the collaboration of a multi-disciplinary team” (Eckert & Clarkson, 2005, p. 5). Beyond the ubiquitous fundamental issues and the sheer size implied in cooperation between humans, the global scale, complexity and creative nature of engineering design processes adds influences to the communication processes. Furthermore, many engineers prefer a visual mode of information processing rather than a verbal mode which could be better supported by current information technology. (Maier & Störrle, 2011)

Challenge: Products and Services

In the last years, the trend can be observed that companies move from selling physical products to offering services and offer product-services systems. This can, for instance, be seen in the aerospace industry in moving from selling gas turbine engines by the unit to providing power-by-the-hour. (Maier & Störrle, 2011) Also in the automotive industry in Germany can be seen as example where different OEMs try to place car-sharing ventures.

Physical nature of engineering artefacts: “The product is the strongest constraint on its own design process” (Eckert & Clarkson, 2005, p. 9). Processes are created to support the development of a product or service and therefore the structure of processes should ideally be closely connected to the product architecture (Sosa et al., 2002). However, the physical nature of many artefacts of engineering design processes adds constraints. For example, combinations

of physical, electrical, mechanical and space restrictions may sometimes be traded-off against each other, while at other times global optimization is required. (Maier & Störrle, 2011)

Economic and market constraints: Engineering design processes are embedded in economic contexts which links to the three sacred cows of engineering design process management: time quality and budget. To achieve the desired product within given time, cost and resource constraints, companies are looking for an appropriate design process. “Tendering agreements define timescales and budgets and often impose harsh penalties for late delivery” (Eckert & Clarkson, 2005, p. 9). (Maier & Störrle, 2011)

Legislative and regulatory constraints: Stringent regulations from different sources apply for many engineering artefacts and need to be addressed within the corresponding engineering design processes. For instance, technical standards imply specific forms of tracing and legislation on safety and environmental certifications influence engineering design processes (e.g. Jarratt et al. (2003) or various ISO, DIN, VDI norms and standards). (Maier & Störrle, 2011)

Importance of characteristics

Additionally to the identification of the characteristics, Maier & Störrle (2011) ran a survey with engineering design process modeling experts from industry and academia to prioritize their importance. Table 2-3 shows the results of the survey: It can be seen that iterative is most strongly supported as important. Together with the interactions of the engineering design process with the organization, other processes and people, complexity is also seen as important characteristic.

Table 2-3 Characteristics of engineering design process in descending order of importance (based on survey of Maier & Störrle (2011))

Mean <i>1: not important; 2: somewhat important; 3: important; 4: very important.</i>	Characteristics of engineering design processes in descending order of importance
3.8	iterative
3.5	process-organization interactions
3.4	complex
3.4	process-process interactions
3.3	people-people interactions
3.1	economic & market constraints
2.9	physical nature of artefact
2.8	legislative & regulatory constraints
2.5	ill-defined

It is arguable if all characteristics are on the same level. For example Roelofsen et al. (2008) and Le et al. (2012) see iterations as result of the process's complexity. Other authors state that it are the interactions and constraints of the process which significantly contribute to its complexity (De Weck et al., 2011; Kortler & Lindemann, 2011; Wynn et al., 2010).

2.2 Engineering design processes as complex systems

Within the engineering discipline, no standardized definition of complexity exists; rather a multitude of specific perspectives can be identified (Piller & Waringer, 1999; Ramasesh & Browning, 2014). Figure 2-1 illustrates common perspectives on complexity in engineering. The system theory perspective usually differentiates between structural and dynamic complexity of systems. While there seems to be agreement on the term structural complexity, there are more detailed differentiations of dynamic complexity (see Geraldi et al. (2011) for a review). This thesis uses the view of Oehmen et al. (2015) to differentiate between structural and dynamic complexity and uncertainty as well human behavior as drivers of complexity.

Other authors distinguish between originating fields of complexity (for example Spur & Eßer (2013)). Oehmen et al. (2015) also differentiate between the complexity of the project and the complexity of its deliverable. The approach of holistic complexity management mentions three management strategies for complexity (Schoeneberg, 2014; Wildemann, 2000). Another perspectives are drivers of complexity which can for example be found in Adam (1998). The mentioned perspectives on complexity are not intended to be exhaustive, but show that complexity has many facets.

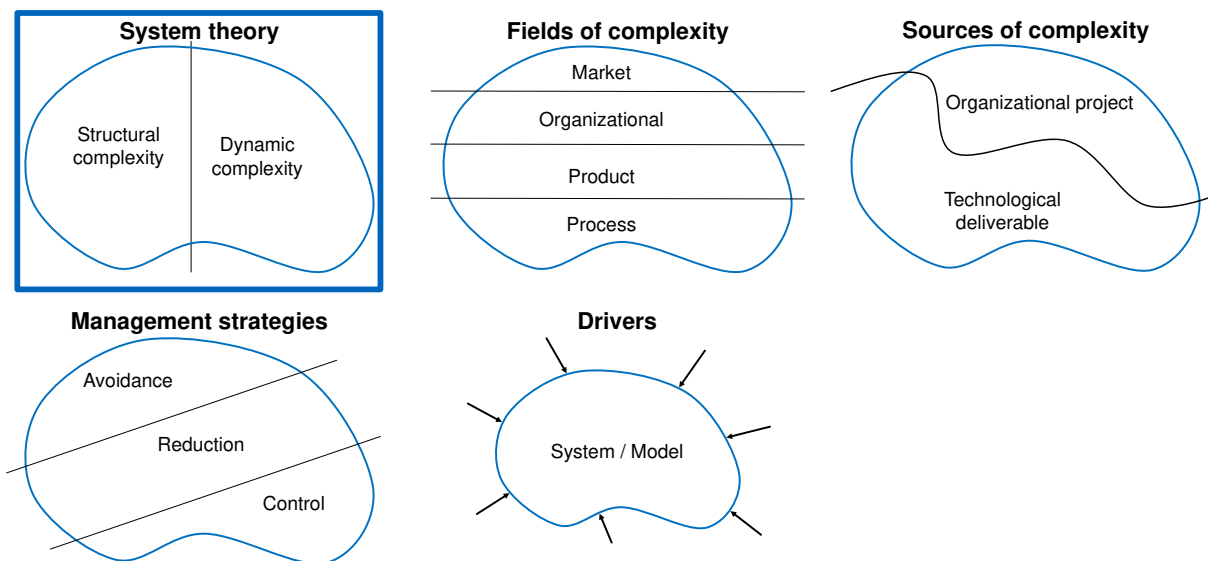


Figure 2-1 Perspectives on complexity (without intending to be exhaustive), with highlighted perspective of this thesis

2.2.1 Terms and definitions

In most cases complexity is understood as system property (Schuh, 2005). Therefore the term system is defined first. There exist several views and definitions of the term system which differ significantly (Bartolomei et al., 2012; R. J. Brooks & Tobias, 1996). An aspect which is common for many definitions is that systems comprise elements (Bartolomei et al., 2012; Wasson, 2006) which are in relationship with each other (Haberfellner et al., 2012; Wasson, 2006). Following Wasson (2006) who suggests to select the best fitting definition for the term system for the application at hand, this thesis uses the definition by Maurer (2007, p. 31):

“A system is created by compatible and interrelated parts that form a system structure, possess individual properties, and contribute to fulfill the system’s purpose. Systems are delimited by a system border and connected to their surroundings by inputs and outputs. Changes to parts of a system can be characterized by dynamic effects and result in a specific system behavior.”

In this thesis complexity is discussed from a systems-oriented perspective. De Weck et al. (2011) state that modern life is governed by engineering systems that fulfil central societal functions, for example modern communication, healthcare, transportation or energy generation and transportation systems. However, these systems are not only of technical nature, they are socio-technical systems where technology and people are intertwined and are dependent on one another. (Oehmen et al., 2015)

De Weck et al. (2011) support the thesis that the key concept in engineering systems is complexity. For defining complexity of engineering systems, they concentrate on two aspects:

“A system is behaviorally complex if its behavior is difficult to predict, analyze, describe, or manage. [...] a system is structurally complex if the number of parts is large and the interconnections between its parts is intricate or hard to describe briefly.” (De Weck et al., 2011, p. 185)

These aspects agree with Oehmen et al. (2015, p. 5) who state that complexity as a property is typically defined by:

- containing multiple parts
- possessing a number of connections between the parts
- exhibiting dynamic interactions between the parts
- the behavior produced as a result of those interactions cannot be explained as the simple sum of the parts (emergent behavior).

The first two of the these bullet points refer to what de Weck et al. (2011) consider structural complexity while point three and four refer to behavioral (dynamic) complexity.

Furthermore de Weck et al. (2011, p. 185) state that:

- “Systems that are structurally complex are usually behaviorally complex.”
- “Systems that have complex behavior need not have complex structure, since we know of relatively simple mechanical systems whose behavior is chaotic, and hence complex.”
- “Most behaviorally complex systems are structurally complex as well.”

Based on these insights de Weck et al. (2011) define complex engineering systems as stated below:

“Complex (engineering) systems are not simply technical in nature, but rely on people and their organizations for the design, manufacturing, and operation of the system, and are influenced by and influence the societal and physical context as well.” (De Weck et al., 2011, p. 185)

An important concept in the context of complexity is complicatedness. In contrast to complexity, complicatedness (or cognitive complexity) is more subjective and observer-dependent. For example, a software application may seem more or less complicated depending

on the simplicity and elegance of its user interface, regardless of the complexity of its underlying code. For engineering design processes, the complicatedness depends on the ability to understand and manage the process. It is influenced by factors such as the intuitiveness of the process structure, organization, and behavior; the ease with which cause and effect relationships can be identified and the experience of the observing persons. Even though complexity and complicatedness may interact and correlate, there is no generalizable causal relationship between them. (Browning & Ramasesh, 2015; Ramasesh & Browning, 2014)

A review on the relation of complexity and complicatedness can be found in Ramasesh & Browning (2014).

Complex systems are developed in engineering design processes which are governed and driven by three key factors according to Oehmen et al. (2015):

- **Technical and organizational complexity:** Parts and interfaces of technical elements of the system, as well as people, their interfaces and relationships to one another have to be managed. Both are tightly coupled, for example, the team structures in engineering often correspond to the system module structure of a product. (Oehmen et al., 2015)
- **Social intricacy of human behavior:** Even though a lot of humans like to think of themselves as rational beings, human behavior is often driven by subconscious thought processes. These processes govern how we face and react to our collective challenges as teams and society as a whole. (Oehmen et al., 2015)
- **Uncertainty of long lifecycles:** Products are part of systems with much longer lifecycles. For example, while a smart phone may have a lifecycle of a few years, it is part of the companies that market them, the communication infrastructure of which they are a part, or the supply chain that extracts and processes the necessary raw materials. Long-term lifecycle considerations have to be part of all of our activities due to the scale that human activity has reached. Among other factors, this increases the uncertainty to which engineering design processes and their outcome are exposed. (Oehmen et al., 2015)

Following the ideas of de Weck et al. (2011), Oehmen et al. (2015) defines the term “complex projects” from an engineering systems perspective. The execution of an engineering design process can be seen as project (Project Management Institute Inc, 2013, p. 3). Additionally, Oehmen et al. (2015) their selves use the example of the development of a new car, which is an engineering design process, to explain why complex projects are complex systems. Consequently, their explanation for complex projects being complex systems can also be adapted to engineering design processes:

Engineering design processes are complex systems because they “[...] are characterized by feedback loops and unforeseen emergent behavior that can spiral out of control, but are fundamentally still tractable by structured (if costly and time consuming) analysis. This domain is inherently knowable encompassing both “known unknowns” and “unknown unknowns”.” (Oehmen et al., 2015)

Based on their definition Oehmen et al. (2015) characterize engineering design processes as possessing

- knowable characteristics (“known unknowns” and “knowable unknown unknowns”)

- discoverable cause-and-effect relationships, which may not be immediately apparent to everyone
- the possibility of more than one right solution.

Facing engineering design processes as complex systems, Oehmen et al. (2015) suggest the managerial strategies of systems-oriented analysis, experimentation, interpretation, and involvement of experts in order to explore different opportunities. Engineering design processes and their environment need to be actively investigated (i.e., analyzed and modeled). It will also be necessary to involve experts in the particular field of the engineering design process for its analysis and management. They will typically provide conflicting advice, which must be discussed and agreed on. (Oehmen et al., 2015)

Managers need to create an environment where new ideas are heard, and that provides space for experimentation to find the best solutions. (Oehmen et al., 2015)

In the following sections the two main aspects of complexity according the systems theory perspective (De Weck et al., 2011) will be discussed: Structural complexity, also called static complexity, which refers to the number and types of elements and their relationships in the system, and dynamic complexity, which refers to the behavior of a complex system (Casti, 1979; Oehmen et al., 2015).

2.2.2 Structural complexity

“Understanding the structure of a system (i.e., its architecture) is a key building block to predicting the system’s behavior.” (Oehmen et al., 2015)

Structural complexity is the most mentioned type of complexity in the literature (Geraldi et al., 2011; Williams, 1999). Most authors determine it by three attributes: size (or number), variety and interdependence (see Geraldi et al. (2011) for a review). According to Maurer (2007), the structure of a system can be seen as the network formed by dependencies between system elements and represents a basic attribute of each system. Structures can be characterized by the specific compilation of relationships between system elements and can be divided into subsets (Maurer, 2007).

Structural complexity of an engineering design process can be illustrated by using stakeholders as example: The process becomes more complex as the number of stakeholders and the differences between the stakeholders increase. If the number of relevant relationships between stakeholders increases, and the types of relationships become more different (e.g., financial flow, information flow, material flow, control flow), the complexity increases too. An internal IT project which improves the communication between first- and second-level supports has low stakeholder complexity. However, a large program that implements a new public health policy has a much more complex stakeholder landscape, thus structural complexity. (Oehmen et al., 2015)

Classic approaches in increasing understanding about a complex system focus on structural complexity. This is based on the procedure of dividing a system into subsystems, noting the relationships between the subsystems that give rise to the system’s behavior, and noting the system’s inputs and outputs. (Browning, 2001)

Once a system structure is sufficiently decomposed, the consequences of the specific system impact and system behavior can be estimated (Lindemann et al., 2005). In other words the knowledge about a system's structural compilation allows for a better prediction of its behavior (Maurer, 2007).

For example, Sharman & Yassine (2004) show that structural attributes contribute to a large extent to system behavior. Also for example Baldwin & Clark (2000) use structural product attributes for characterizing system behavior. Baldwin & Clark (2000) mention that a minimization of dependencies between system elements within the product architecture also improves the future development of complex systems. Thus, the structural complexity of the product influences its development process. (Maurer, 2007)

Additionally, adaptation time and costs can be reduced and system reliability can be increased if adequate system structures are actively designed (Lindemann et al., 2005).

2.2.3 Dynamic complexity

“Complex systems can result in emergent phenomena that could not be predicted by the characteristics of the components parts or subsystems. This is often true of systems whose subsystems have a degree of autonomy and their own objectives.” (Rouse, 2007)

The analysis of the system structure, thus of the structural complexity as shown in the previous section, uses a static snap-shot of the system in a particular point in time (Lichtenberg et al., 2013; Oehmen et al., 2015). Furthermore, it is also possible to analyze and understand the system in terms of its behavior and how it changes over time (Oehmen et al., 2015). Oehmen et al. (2015) state that while the idea of doing so is relatively straightforward, the actual analysis and interpretation are more difficult.

However, some of the most important aspects of complex projects (such as engineering design processes) relate to their dynamic nature (Geraldi et al., 2011; Kreimeyer, 2009; Maurer, 2007; Oehmen et al., 2015; Xia & Lee, 2005). Typical aspects are process duration, development and change of stakeholder requirements over time, change of staffing levels and possible emerging behaviors such as organizational resistance to a change process (Oehmen et al., 2015).

Roth & Senge (1996) characterize dynamic complexity as the extent to which cause and effect are distant in time and space. In situations of high dynamic complexity, the causes of problems cannot be easily determined by first-hand experience, and few, if any, of the actors in the system may have a substantive understanding of the causes of problems (Roth & Senge, 1996).

The dynamics of complex systems results from cause-and-effect chains which are formed by the present dependencies within and between market, organizational, process and product complexity and become ambiguous and nonlinear (Maurer, 2007; Xia & Lee, 2005). Xia & Lee (2005) especially note that dynamic complexity is caused by changes in organizational and technological project environments. These changes may result from either the stochastic nature of the environment or a lack of information and knowledge about the project environment (Xia & Lee, 2005). Dynamic complexity is characterized by feedback loops, and non-linear and emergent (both planned and unforeseen) behavior (Oehmen et al., 2015). Xia & Lee (2005) state that the performance of dynamically complex tasks requires knowledge about how the

structural complexity (referred to as component and coordinative complexities by the authors) changes over time.

Based on the statements found in literature it can be concluded that for this thesis **dynamic complexity**:

- results in emergent phenomena
- is caused by changes in organizational and technological environments originating from the environments' stochastic nature or lack of information about the environments
- is the extent to which cause and effect are distant in time and space and their chains are ambiguous and nonlinear
- is more difficult to analyze and interpret than structural complexity
- its management requires knowledge about how structural complexity changes over time.

2.2.4 The relationship of complexity, uncertainty and human behavior

“Complexity, per se, in engineering projects [...] is determined by the technical complexity of the product system being developed, as well as the organizational complexity of the project [...] developing the product. The two systems are tightly coupled, increasing the structural and dynamic complexity. Uncertainty of project execution, as well as built-in behavioral patterns of humans, increases the complexity faced in the projects.” (Oehmen et al., 2015)

Structural complexity can be understood as the complexity of how an engineering design process is built, while dynamic complexity can be understood as how it behaves. However, this does not explain where complexity comes from. It can be argued that there are three major drivers behind complexity: Complexity “per se,” human behavior and uncertainty. (Oehmen et al., 2015)

Complexity “per se”

During engineering design processes two types of complexity need to be managed: First, the organizational complexity of the process as project itself (**organizational project complexity**). This includes the structure of the process organization (i.e., stakeholders and their relationships), as well as its behavior (i.e., processes). Second, the complexity of the main process deliverable (**technological deliverable complexity**), thus the product, system, or service to be developed, needs to be managed. That consists of its structure (i.e., product architecture) and “behavior” (e.g., properties along its lifecycle, such as environmental impact, maintainability, or profitability). Project complexity and deliverable complexity are closely related to one another. (Oehmen et al., 2015; Xia & Lee, 2004)

Typical relationships according to Oehmen et al. (2015) are:

- The architecture of a deliverable typically defines a significant part of the structure of engineering design process. If the technical architecture is flawed, the integration step between work packages will be difficult to impossible.
- The nature of a deliverable determines the key structure of the engineering design process. For example, in order to execute an agile, incremental process execution, the

deliverable must be “creatable” and testable in incremental units (which may be difficult for a fighter jet). Hence, an Engineering Design process has to reflect the realities of the nature of the deliverable, while taking full advantage of developing capabilities.

- The number and variety of stakeholders significantly influence requirements and scope, as well as requirements and scope creep. The structural complexity of engineering design processes has to be reduced (i.e., the number and diversity of key stakeholders in the governance structure) in order to limit the complexity of the resulting solution.

Uncertainty

“We are bad at intuitively dealing with uncertainty and instinctively avoid and distort its perception. Additionally, the concept of ambiguity (unknown uncertainty) describes how two individuals can derive completely different conclusions from the same factual, uncertainty-related information.” (Oehmen et al., 2015)

A large number of problems in the context of engineering design require to make decisions in the presence of uncertainty (X. Yin, 2009). Especially the early phase of engineering design processes is predominated by a high amount of uncertainty (Lévárdy & Browning, 2009; Wiebel et al., 2013). The early phase thereby includes the phases of planning, task clarification and conceptual design and is elaborated in detail in section 2.4.

Uncertainty can thereby be characterized as a lack of knowledge about a problem at the time of making a decision affecting its solution. It can be distinguished between known uncertainty as “knowing what you don’t know” and unknown uncertainty (ambiguity) as “not knowing what you don’t know” (Sarbacker & Ishii, 1997; Schrader et al., 1993). (Lévárdy & Browning, 2009)

Unknown uncertainty can only be mitigated by general aspects of conservative design, thus an intensive search for uncertainties should be conducted, which should be taken into account during engineering design processes actively and in a target-oriented way. (Schrieverhoff, 2014)

Due to the strong presence of unknown uncertainty in engineering design processes, the traditional methods and tools of project management provide less value. It is important that the project managers’ and participants’ attitudes and aptitudes must change (De Neufville, 2004; Loch et al., 2006; Wiebel et al., 2013) and presumably also their models, methods, and tools. (Lévárdy & Browning, 2009)

While unknown uncertainty is a source of risk for engineering design processes, it can also bring opportunities to organizations which are capable of effectively sensing the endogenous and exogenous changes and responding to them efficiently by adapting to the changed conditions (Haeckel, 1999). (Lévárdy & Browning, 2009)

Engineering design processes which are capable of coevolving with their environments and dynamic stakeholder needs can profit from the accelerating pace of changing market needs (Dougherty, 2001; Eisenhardt & Tabrizi, 1995). Due to the increasing acknowledgement of uncertainties consideration is given to nontraditional approaches to project management such as *extreme* (Beck, 1999), *adaptive* (James a Highsmith & Highsmith, 2013), *flexible* (MacCormack et al., 2001), *response-able* (Dove, 2002), *lean* (Poppendieck & Poppendieck,

2003), *agile* (Jim Highsmith, 2004) etc.. However, they may not apply as readily for complex hardware systems (Boehm & Turner, 2003). (Lévárdy & Browning, 2009)

Lévárdy & Browning (2009) argue that the assumptions underpinning the conventional models and tools for engineering design process planning and control do not always hold (Koskela & Howell, 2008; Winter et al., 2006), and that these models could extend their capabilities by accommodating more dynamism and flexibility. (Lévárdy & Browning, 2009)

Oehmen et al. (2015) argue that current strategies for managing uncertainty do not justice to the types of uncertainty and variety of information quality regarding the uncertainty we encounter in complex projects. According to the authors, managers tend not to differentiate between aleatoric uncertainty (uncertainty caused by a lack of knowledge, such as the performance of a particular technology under field conditions) and epistemic uncertainty (fundamentally unknowable outcome, such as the result of a throw of dice). This results in sub-optimal performance of complex risk situations. (Oehmen et al., 2015)

Oehmen et al. (2015) also argue that managers traditionally favor a probabilistic view of uncertainty, which implies that a minimum amount of knowledge regarding an uncertainty is necessary in order to arrive at a meaningful estimation of the related probability. In complex projects such as engineering design processes, this quality of information is often unavailable, but non-probabilistic assessments of uncertainty (which have lower information requirements) are not typically used. Leading to inaccurate risk estimates and lower planning quality, the perceived complexity of engineering design processes increases. (Oehmen et al., 2015)

Human Behavior

“Our human minds have limited capacity and cannot intuitively comprehend even mildly complex systems. To make matters worse, our intuitive decision-making rules conspire against us in the face of complexity to produce catastrophic results. We are also incapable of intuitively perceiving the most fundamental aspects of complexity, such as feedback loops, exponential growth, or low probabilities.” (Oehmen et al., 2015)

The ability to make sound decisions in the face of complexity is limited by three factors:

- The imperfect access to information and thus, the complete and accurate representation of the full current state (Box & Draper, 1987; Oehmen et al., 2015; Sterman, 2000; von Neuman, 1947).
- The cognitive limitations of humans such as the number of factors that can be considered in parallel, the amount of information that can be processed and the speed of processing (Bood & Postma, 1997; Lorenz, 2008; Oehmen et al., 2015).
- The limited time to make a decision (Oehmen et al., 2015; Rahmandad & Sterman, 2008).

Especially in situations with high complexity, the limited attention capability of decision makers is faced with a very high (possibly endless) amount of information which needs to be understood and processed. Given this fact, the fundamental challenge arises of finding the right way of compressing complexity without sacrificing key aspects that are relevant for decision making. Additionally, and as reaction to the factors stated above, humans have developed subconscious or built-in decision-making models and rules. However, these capabilities

developed over thousands of years, are of limited help when managing complexity. (Oehmen et al., 2015)

Typical challenges of subconscious and built-in decision-making models and rules according to Oehmen et al. (2015) are:

- Number and diversity of items: While complex systems often consist of a high amount of elements, humans are only able to keep in mind about five to seven elements at any given time.
- Dynamic behavior and change: Changes are subconsciously extrapolated as linear.
- Cherry picking: Perceived complexity is often reduced by selective attention. In an engineering design process, this leads to a lack of appreciation of complexity: Decision makers may get stuck in a set frame of mind.
- Overconfidence and optimism bias: For example people usually prefer to start execution with “easy” tasks. This may lead to overly optimistic assessments of cost and schedule requirements, and chances of success. As shown in various studies, over 90% of drivers believe their driving skills are above average.
- Hindsight bias (knew-it-all-along effect): After an event has occurred, humans tend to see the event as having been predictable, despite there has been little or no objective basis for predicting it (Hoffrage & Pohl, 2010; Roese & Vohs, 2012). For complex situations, this creates a false trust in the quality of our “intuition” to foresee problems and choose the right solutions.
- Anchoring bias: Humans tend to rely too heavily on the first piece of information offered (the "anchor") when making decisions (Investopedia, 2016). Untrained estimations often completely ignore even the limited information that may be available. They can be worse than using random numbers.

(Oehmen et al., 2015)

Complexity can be seen as a multi-dimensional concept. The structural and dynamic complexity of engineering design processes is driven by the organizational complexity of the engineering design process as project and the technological complexity of its deliverable. Furthermore complexity is driven by the prevailing uncertainty and human behavioral models and rules which are often of limited help when managing complexity.

2.3 Modeling of engineering design processes

While the previous section elaborated on engineering design processes as complex systems with its dimensions and drivers, this section gives an overview of existing modeling approaches. Modeling is a classic approach to address and understand a complex reality (Browning, 2002). In comparison to the object being modeled, a model represents a target-oriented, simplified formation analogous to the original, which permits deriving conclusions based on the original object (Lindemann, 2007). Following a classification subsection, the systems theory perspective with its dimensions of structural and dynamic complexity will be maintained to detail existing approaches.

Process modeling supports the management of engineering design processes to cope with complexity and uncertainties emerging from both internal and external factors (Browning et al.,

2006; Smith & Morrow, 1999). Process modeling is typically employed to capture characteristics of the process structure and the process environment, explore potential interactions between these two levels and potential influence of project management on the process behavior. This helps to create a better understanding of process behavior and how the process structure, environment and management might influence this behavior. A better understanding of process behavior and its influences improves decision making. (Le, 2013)

As for other modeling activities, also during process modeling trade-offs between expressive power and completeness on the one hand and simplicity of the models on the other have to be made (Gemino & Wand, 2004). Various process modeling techniques with different abstraction levels and perspectives have been developed; their insights may be different but complementary (Le, 2013). Browning (2009) proposes the development of a process architecture framework to help manage the complexity in engineered systems: It provides a portfolio of views of a complex system, each of which describes it partially and in a meaningful format to its users and their particular needs (Browning, 2009).

Typical purposes of process modeling are related to their visualization, planning, execution, control or development (Browning & Ramasesh, 2007). There is no commonly agreed classification of models of engineering design processes. However, most common modeling techniques found in the Engineering Design and corresponding literature can be classified into models of the structural and dynamic complexity (e.g., (Browning & Ramasesh, 2007; Browning, 2002; Browning et al., 2006; Smith & Morrow, 1999; Wynn et al., 2007).

2.3.1 Classes of engineering design process models

O'Donovan (2004) suggests to classify engineering design process models along two axes: generic against specific models and descriptive against prescriptive models. Using this classification he identifies four rough groups of models which are illustrated in Figure 2-2:

- prescriptive/descriptive generic models
- descriptive generic models
- product-based models
- modeling frameworks and instances.

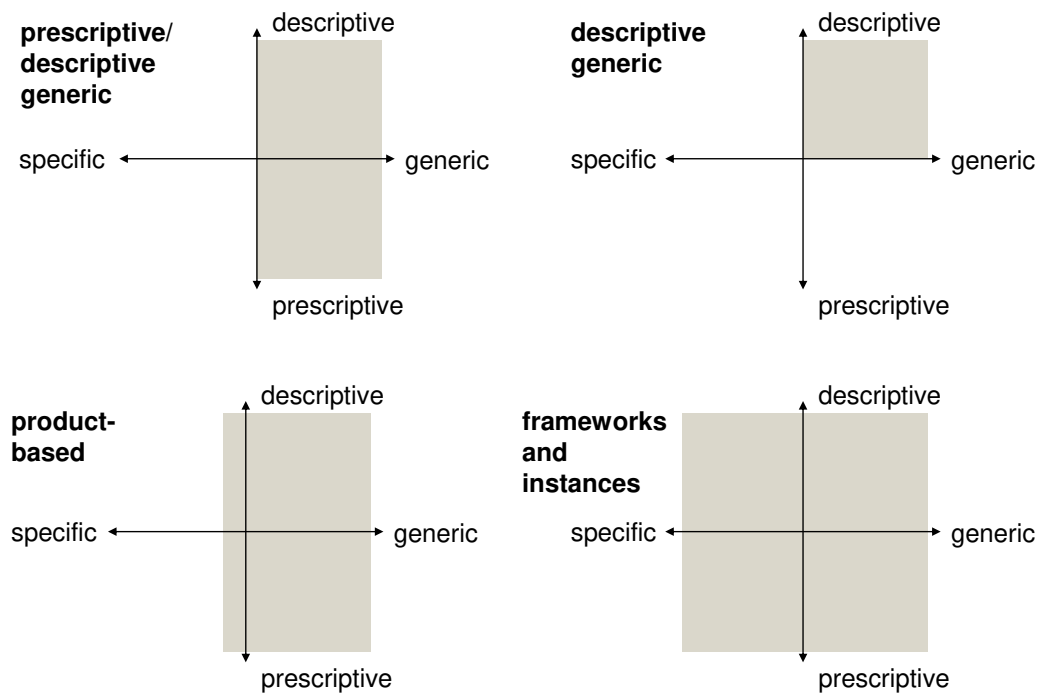


Figure 2-2 Classification of engineering design process models according to O'Donovan (2004)

Prescriptive/descriptive generic models: According to O'Donovan (2004) the primary purpose of these models is teaching. By highlighting the types of information, specifications, concepts, detailed designs, they provide a rough checklist of what needs to be done in early, middle and closing stages of an engineering design process and many concepts are of value in original design. They can also be interpreted as descriptions of an ideal design process, or as prescriptions of best-practice in design. A typical example is the design process model of Pahl & Beitz (1996). (O'Donovan, 2004)

Descriptive generic models: Descriptive generic models are solely intended for the description of the design process. They cannot be used as a basis for imposing specific conditions on a process, because the reflected patterns of activity include a random element (for example Ullman et al. (1988)), or because the patterns are not a complete generative set of rules for creating a process (for example Chandrasekaran (1990)). Descriptive generic models are framed to deliver insights into the nature of design such as the sub-activities which take place within them or the arrangement of activities. Descriptive generic models often provide a basis on which computational methods are developed. (O'Donovan, 2004)

Product-based models: There are models of engineering design processes which are closely connected to specific products such as ship design parameters (Evans, 1959) or software engineering processes (Muench, 1994). Also models exist that are tied to classes of products at a higher level of detail such as the designer support systems (Duffy & MacCallum, 1989) and the conceptual design of ships and jet engines (Jarrett, 2001). (O'Donovan, 2004)

Frameworks and instances: Models of this class bridge the classes of specific and generic, and the descriptive and prescriptive. A modeling framework can be seen as generic approach which may be applied to modeling any situation within its scope, but which in itself provides little specific guidance or insight. O'Donovan (2004) uses the picture of a sandbox which is the

framework, while the instances are the sandcastles built within it. As the properties of the sand limit the forms that can be created, so a framework places constraints on the features of models that can be built within it, too. Instances can be seen as descriptive and prescriptive: While its initial construction is usually a description of a engineering design process, which can be observed directly, inferred or a statement of intended actions. However, once created the instance may be used for prescription, either to run a project in a similar way or the model may be manipulated to obtain for example an optimized process. (O'Donovan, 2004)

Typical frameworks for engineering design process modeling are (without the intent to be exhaustive):

- IDEF
- Petri Nets
- DSM
- MDM
- System Dynamics
- UML

2.3.2 Modeling of structural complexity

There are different dependency modeling methods for process models. Kreimeyer (2009) cites Belhe & Kusiak (1996) for an overview. Primarily graph theory provides the computational approaches to analyze structures and their subsets. Additionally to hierarchically-ordered structures, the composition of peer linkages can also be characterized by the formation of specific subsets (Lindemann et al., 2005; Wasson, 2006). For example, Foulds (2012) and Gross & Yellen (2005) describe the discipline of graph theory, which provides mathematical fundamentals suitable for structural analysis. (Maurer, 2007)

The primary target of structure considerations is a better understanding of the complex system at hand by representing the entities involved, their relationships, and the quality of their interactions (Browning, 2001; Kreimeyer, 2009). According to Maurer (2007) the two major purposes of modeling structural complexity are:

- information visualization
- application of computational approaches and algorithms.

In engineering design there are two main analysis directions based on the system structure: the analysis of either product structures, also referred to as product architectures and the analysis of process structures, thus process models.

A major concept of product structure analysis is modularity which is for example elaborately discussed by Ulrich & Eppinger (2003). However, this section focuses on modeling the structural complexity of processes, thus process models.

Process models are used for a variety of purposes which coincide with the different goals of process management. Each individual process model depicts a part of the structure of a process. Browning (2009) assesses different process models as to their focus and their different stakeholders to compare what model is made for what purpose. As result, Browning (2009) states that while every model in his review is made for a different purpose, many models convey

similar information. Numerous methodologies for process modeling are available for different needs, and non-exhausted lists and comparisons are for example provided by Baumberger (Baumberger, 2007), Browning (2009), Karniel & Reich (2009) and Kreimeyer (2009). Browning & Ramasesh (2007) review the literature of network-like process models. They find that the interaction of tasks and their impact on overall process improvement can be found in all existing models, yet needs more focus. (Kreimeyer, 2009)

Kreimeyer (2009) concludes that all process models contain aspects of the structure of a system as they all consist of entities and relationships. Within some models the structure is specified very strictly, while others leave more room to adapt the model. (Kreimeyer, 2009)

Engineering design processes can be seen as networks of multiple domains that coexist to enable the development of products. Each of these domains is networked in itself in many different ways and domains are also internally linked and coupled among each other, as well as the behavior of the engineering design process to a large extent depends on this network structure. The communication among organizational units can be seen as example: It is only possible if units are related to each other. Consequently the specific setup of entities and their relations would constitute the value of the actual process. (Kreimeyer, 2009)

Kreimeyer (2009) argues that to model the structure of engineering design processes, they should be understood as multi-layered networks. To analyze such processes it is important to select and relate all relevant domains in an integrated manner which simultaneously enables and facilitates systematic and comprehensive analysis. Kreimeyer (2009) further suggests using the modeling technique of the Multiple-Domain Matrix (MDM) for representing and manipulating network structures consisting of different domains and relationship types. The concept of MDMs was originally developed as part of the Structural Complexity Management methodology (Lindemann et al., 2009).

Based on the observation of Belhe & Kusiak (1996) that all major models can be converted among each other using adjacency matrices, Kreimeyer (2009) identifies adjacency matrices as similar to DSM (Design Structure Matrix) respectively MDMs.

MDMs allow to represent multiple network structures, both within a single domain (e.g., process steps) and across domains (e.g., the attribution of organizational units to individual process steps). MDMs also allow to capture different relationship types that coexist concurrently which makes them an ideal tool for modeling the structure of engineering design processes. Figure 2-3 visualizes the concept. (Kreimeyer, 2009; Maurer, 2007)

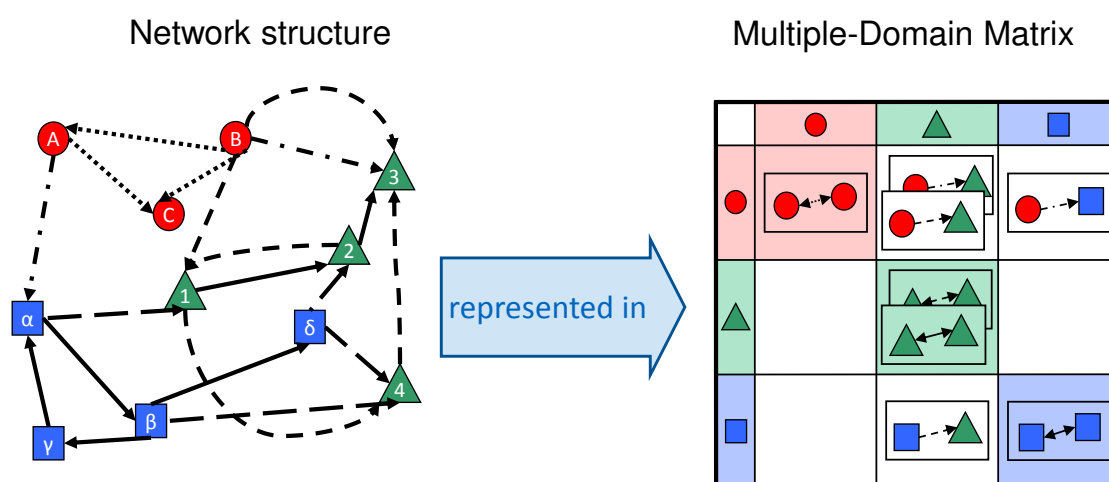


Figure 2-3 Capturing of different relationship types within a MDM (Maurer, 2007)

The MDM combines the possibilities of the Design Structure Matrix (DSM) (Steward, 1981) and the Domain Mapping Matrix (DMM) (Danilovic & Browning, 2004). DSM and DMM modeling techniques are applied for modeling and analyzing system structures in a multitude of different projects in which elements of different domains are focused (for an overview see Browning (2001)). The MDM supports the handling of multiple-domain systems as it allow to represent multiple network structures, both within a single domain and across domains (Kreimeyer, 2009; Maurer, 2007). Thereby, domains represent the classification of the elements in groups (Lindemann et al., 2009). Examples of domains are process steps and organizational units and single elements represent specific instances of these groups. The MDM also allows for capturing different relationship types that coexist concurrently. The MDM is the framework of the Structural Complexity Management methodology of Lindemann et al. (2009). The methodology provides a five-phase procedure that supports users in system definition, information acquisition, deduction of indirect dependencies, structure analysis, and the application on the product design (see Figure 2-4).

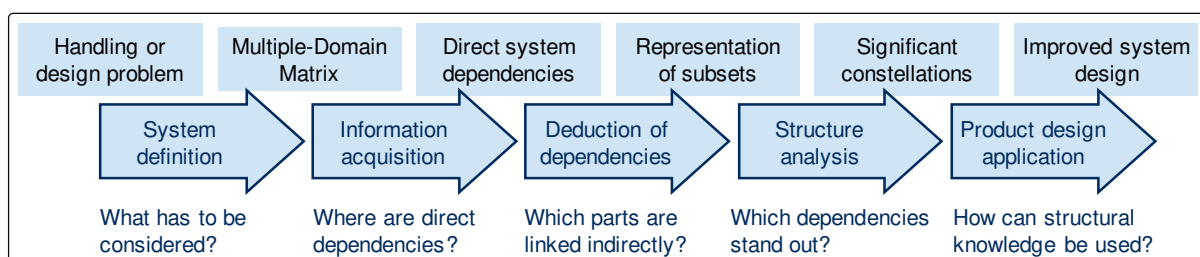


Figure 2-4 Phases of the Structural Complexity Management methodology (Kasperek et al., 2013), adapted from Lindemann et al. (2009)

For the deduction of indirect dependencies and structure analysis, algorithms for calculating DSMs from DMMs are used. The analyses are computed in a MDM which consists of at least two, but theoretically up to an infinite number of domains. The domains (and with it the granularity of the model) are chosen either according to the intended results of the later analysis or according to the existing information sources (Lindemann et al., 2009).

Whereas the methodology of Structural Complexity Management has been created in the context of engineering design of products, it has been enlarged to several other areas of research recently, e.g. process improvement (Kreimeyer, 2009), knowledge transfer (Maurer, 2011) or security management (Maurer et al., 2009). Software support is available for supporting the acquisition, representation and analysis of system structures.

Based on a literature review, Kreimeyer (2009, p. 112) identifies domains, describing what types of entities are common to process modeling, and the relationship types, describing how the domains are commonly related which together form his “Meta-MDM with domains and relationship types suited for most modeling and analysis purposes”. The six domains which were identified as most common and represent the usual domains found in process modeling can be found in Table 2-4.

Table 2-4 Most common domains in the context of engineering design processes, identified by Kreimeyer (2009)

Domain	Description
Task	The task domain collects all entities that describe the execution of work done in the process; further terms are: Function, method, action, activity, unit of behavior, gateway, transition and work package.
Artifact	The artifact domain regroups all objects that are intermediate and final input and output objects in the process. Some modeling languages differentiate between artifacts focused on value generation and on process control; however, here, both kinds are included. Further terms are: Input / output, object, product, data, parameter, and information.
Event	This domain addresses non-persistent occurrences in time that present a certain status or progress. Further terms are: Message, order, initial / final node, label, and place.
Organizational Unit	The organizational unit domain contains all human resources in their respective ordering; further terms are: Staff, responsibility, team, pool, lane, actors, roles, and committee.
Resource	The resource domain is intended for all non-human resources necessary to enable the process execution, such as IT-systems, equipment or knowledge. Further terms are: Attribute, mechanism, method, pool, and lane.
Time	The time domain addresses persistent time issues such as the start time of a process step or milestones in the process. Further terms are: Attribute, duration, starting time, end time, average time, milestone, and phase.

The resulting Meta-MDM, as illustrated in Figure 2-5, is intended as a modeling scheme which is capable to describe relevant aspects of structural modeling and a goal-oriented process analysis. It provides orientation when modeling a structural process model and it systematizes and collects relevant domains and relationship types and puts these into a common framework.

According to Kreimeyer (2009) the MDM metamodel serves the following purposes:

- to be a basis for discussion with experts when collecting and interpreting data
- to guideline which structural elements and interdependencies to consider
- to ease the later transformation of the MDMs into System Dynamics models of engineering design processes
- to support the clear representation of underlying modeling assumptions of the System Dynamics model.

	Process step	Artifact	Event	Org. Unit	Resource	Time
Process step	<ul style="list-style-type: none"> precedes temporally precedes logically 	<ul style="list-style-type: none"> has output of processes controls changes 	<ul style="list-style-type: none"> generates processes ends in 	<ul style="list-style-type: none"> belongs to 	<ul style="list-style-type: none"> requires processes 	<ul style="list-style-type: none"> is active at is finished at has duration of starts at
Artifact	<ul style="list-style-type: none"> is input for supports controls starts 	<ul style="list-style-type: none"> transits into is used to create is in conflict with 	<ul style="list-style-type: none"> sends is created at is needed at 	<ul style="list-style-type: none"> belongs to 	<ul style="list-style-type: none"> is processed by is stored in 	<ul style="list-style-type: none"> is created at is needed at
Event	<ul style="list-style-type: none"> starts controls 	<ul style="list-style-type: none"> produces 		<ul style="list-style-type: none"> occurs in 		
Org. Unit	<ul style="list-style-type: none"> is responsible for is necessary for supports is involved in takes decisions about has to be informed about processes data for 	<ul style="list-style-type: none"> has responsibility for uses creates / deletes / updates is authorized to read 		<ul style="list-style-type: none"> communicates with belongs to is formed by is head of is responsible for takes role as 	<ul style="list-style-type: none"> uses 	<ul style="list-style-type: none"> is active during is finished at starts at
Resource	<ul style="list-style-type: none"> is required for supports is occupied by 	<ul style="list-style-type: none"> processes creates emits stores transmits 		<ul style="list-style-type: none"> transmits message to 	<ul style="list-style-type: none"> transmits message to 	<ul style="list-style-type: none"> is available at is available during
Time		<ul style="list-style-type: none"> is attribute to 				

Figure 2-5

Meta-MDM with domains and relationship types suited for most modeling and analysis purposes of Kreimeyer (2009)

2.3.3 Modeling of dynamic complexity

Engineering design processes are characterized by their dynamic and creative behavior and their unpredictable results (Schlick et al., 2013; Wynn et al., 2003). Because of the uncertainty within these processes, disruptions occur and affect the process flow. If the system's state is time-varying, simulations are usually chosen for modeling the dynamic behavior of the system (Suss & Thomson, 2012).

According to Moon (2015) simulation is a kind of modeling, but refers to a group of methods that imitate the behaviors and characteristics of real systems, normally on a computer. Typical uses of simulation are:

- to develop a better understanding and gain insights of a system
- to compare various plans and scenarios before implementation
- to predict behaviors of a system
- to support decision-making processes
- to develop new tools for investigation
- for training.

(Moon, 2015)

There exist numerous simulation methods. However, this section focuses on the three major methods for the modeling of dynamic complexity:

- Agent-based Modeling (Gilbert, 2008)
- Discrete-event Modeling (Law, 2014)
- System Dynamics (Sterman, 2000).

Agent-based Modeling is a simulation method in which agents are modeled to interact with each other and their environment. The emerging behaviors, patterns, and structures from such interactions over time are results which are derived by interpreting each agent as an individual entity possessing its own intelligence, memory and rules. Thereby, each individual agent makes decisions based on what he perceives from other agents and his environment. The basic idea of Agent-based Modeling is adopting a bottom-up approach starting from individual agents to model complex systems. (Gilbert, 2008; Moon, 2015)

For example, Lévárdy & Browning (2009) developed an Agent-based adaptive process model to product development project management. Garcia (2005) gives an overview of applications of Agent-based models in the context of engineering design and highlights the following areas of application: diffusion of innovations, organizational strategy, and knowledge and information flows.

The basic mechanism of Discrete-event Modeling is that a system's state variables change only at discrete and separate points in time. The only instances where the state of the system changes are events occurring in those points in time. Discrete-event Modeling typically models a complex system as an ordered sequence of events. However, even complicated sequences and hierarchical structures can be employed. Additionally, uncertainties associated with events can be modeled explicitly and their collective consequences in the system are analyzed statistically. (Moon, 2015)

Cho & Eppinger (2005) expand DSMs by Discrete-event approaches for the field of complex engineering design processes. However, their approach cannot simulate influences of uncertainty on the process duration. Meier et al. (2015) build a rich model that accounts for activity overlapping, crashing, and iteration in product development project activity networks.

System Dynamics (SD) is a type of simulation where a system's state variables change continuously over time. The continuous changes in state variables over time are typically expressed by equations (Moon, 2015). Conceptually System Dynamics models complex systems incorporating three elements:

- A stock that is a reservoir for a resource.
- A flow that adjusts the level of stock through inbound flows and outbound flows.
- A link between a stock and a flow.

(Moon, 2015)

In contrast to Agent-based Modeling, System Dynamics adopts a top-down approach, conceptualizing a complex system at a more aggregate level (Moon, 2015).

System Dynamics models describe dependencies in a dynamic system by using single or multiple feedback loops (Rannacher et al., 2013). Thereby they demonstrate how unappreciated causal relationships, dynamic complexity, and structural delays can lead to counter-intuitive outcomes (Tignor, 2004). Additionally, SD accommodates “soft factors” such as motivation and perceptions so that management can better understand problem spaces (Tignor, 2004).

Simulation with System Dynamics models is used for learning about the dynamic complexity of systems, for the identification of optimal policies in existing systems, and for improvement of system behavior through parameter or structural changes. The method has been applied to a wide range of domains, from the management of production distribution systems to the management of ecosystems. (Pfahl & Lebsanft, 1999)

Referring to engineering design processes, Cooper (1980) developed the so called rework cycle to model the typical iterations of engineering design. The construct was picked up by Ford & Sterman (1998a) and others and further developed. It is the core concept of most developed System Dynamics engineering design process models in literature. Schmidt et al. (2015) give an overview of existing variants of the rework cycle in the context of System Dynamics modeling of engineering design processes.

The use of System Dynamics provides a way to simulate the dynamic complexity and thus, enables to foresee the process behavior. Further knowledge of the process flow allows the managers to improve their planning and thus save company money and time. For this reason, many researchers worked together with companies (D’Avino et al., 2005; Le et al., 2012; Lin et al., 2008; Lyneis et al., 2001) and created System Dynamics models for simulating engineering design processes with rework cycles.

In order to best reproduce the process behavior, modelers include certain features in their models to reflect special process characteristics. A detailed review of these features is given by the *Adaption scheme for customizing rework cycles* which is part of the tools and methods developed to support the framework for structure-based System Dynamics Analysis of Engineering Design Processes.

2.4 Early phase of engineering design processes

Many authors have proposed models of engineering design processes that essentially include a linear sequence (Lessio, 2011). One widely known and cited model is the design process model of Pahl & Beitz (1996). They present a systematic approach that comprises the four phases of planning and clarifying the task, conceptual design, embodiment design and detail design.

Referring to the engineering design process model of Pahl & Beitz (1996), within this thesis the early phase of engineering design processes includes the phases of planning and clarifying the task as well as conceptual design. This aligns with the definitions of Tatarczyk (2009) and Murphy & Kumar (1997).

The phases of planning and clarifying the task consist of all activities prior to the explicit generation of solutions. Within these phases the needs are analyzed and the problem, goal and requirements are formulated. (Blessing, 1994)

Based on a literature review Blessing (1994) highlights the importance of the planning and task clarification phase, in particular, the necessity of an unambiguous specification, which should be as complete as possible. Within this phase the product idea arises, strategic directions are being formulated and the design specified to a degree that will already determine the likelihood of a product's success (Roozenburg & Eekels, 1995). (Pohlmeyer, 2011)

Conceptual design is the phase in which principle solutions or concepts for the product and its main elements are generated based on the functions the product has to fulfil. Hubka (1984) considers the conceptual design phase to be among the most demanding phases in design work and indeed the whole of engineering. (Blessing, 1994)

The early phase of engineering design processes is predominated by a high amount of uncertainty (Lévárdy & Browning, 2009; Wiebel et al., 2013).

However, the decisions which are made in the early stages of engineering design processes will affect all following decisions, activities, and results. Even though they can also be adapted and modified later in the process as information increases, development time and the number of cost-intensive modifications are reduced if the task has been sufficiently clarified and requirements were correctly and sufficiently specified prior to the subsequent design phases (Pahl, Beitz, Feldhusen, & Grothe, 2007). (Pohlmeyer, 2011)

Modifications can still be made relatively easy in the early stage. Any later modifications that will be necessary will lead to increases in cost and time (Bias & Mayhew, 2005; VDI, 1994). The ability to influence the outcome decreases as the development progresses (Wheelwright & Clark, 1992). The use of a systematic design methodology which is an early investment of a considerable amount of time, is expected to decrease the overall project time in the end (Pahl, Beitz, Feldhusen, & Grothe, 2007). (Pohlmeyer, 2011)

Linking to the objective of this thesis, understanding the intrinsic complexity with its dimensions and their interactions, decreasing the perceived complicatedness and uncovering risks during the early phase is a strong lever for the overall process, as its structure can still be adapted during this phase.

2.5 Existing models combining structural and dynamic complexity of engineering design processes

“Effective decision making and learning in a world of growing dynamic complexity requires us to become systems thinkers to expand the boundaries of our mental models and develop tools to understand how the structure of complex systems creates their behavior.” (Sterman, 2000)

The structure of complex systems such as engineering design processes is the foundation of their behavior, i.e. its dynamics (Flurschein, 1977; Kreimeyer, 2009; Rechten, 1991; Wasson, 2006). This section reviews existing models combining structural and dynamic complexity of engineering design processes.

To identify existing combinations of structural and behavioral models in Engineering Design a literature study was conducted. The research platform Thomson Reuters Web of Knowledge was used to identify an initial set of literature, which was then further extended by using Google Scholar for cross referencing.

Within a first step all articles were searched having keywords from the structural modeling context (structure or dependency) as well as from the behavioral context (behavior) and from the engineering process context (engineering and process). As 33.623 articles were identified on the research platform, this initial set was further refined: For each of the three search domains indicated in Table 2-5 more detailed keywords were defined. As a second step within the 33623 articles all articles containing at least one of the keywords were identified. For the remaining 916 articles duplicates in the database were deleted and the articles' titles and abstracts were screened to further narrow the relevant literature. Overall 66 accessible articles were identified by the literature search. These 66 articles and their cross references were further examined to identify existing combinations of structural models and behavioral models in Engineering Design. The complete list of the considered 66 articles can be found in the appendix 9.1.

Table 2-5 Detailed keywords for the three search domains

Search domain	Keywords
System Structure	Design Structure Matrix, DSM, Domain Mapping Matrix, DMM, Multiple-Domain Matrix, MDM
System Behavior	System Dynamics, Discrete Event Simulation, Agent-based Modeling, Multi-agent System, Structural Complexity Management, Systems Engineering
Engineering Design	Engineering Design, Product Development

In the following, based on a review of the identified articles, a summary and interpretation in the context of this thesis of the most relevant articles found is presented.

Identifying Controlling features of Engineering Design (R. P. Smith & Eppinger, 1997)

Within their work, Smith & Eppinger (1997) identify controlling features of Engineering Design iteration by using DSMs. They claim that the complex set of relationships among a large number of coupled problems leads to iterations in engineering design. They suggest the DSM as useful to identify where iterations will occur. In their work, they developed the so-called work transformation matrix to predict coupled features which will require many iterations. The aim of this model, which uses linear systems theory, is to describe the underlying behavior of

complex design iteration processes. The authors state that models of iteration provide essential understanding of the effects of complex relationships within development processes and help identify controlling features, which is an important step of improving and shortening development time.

Interpretation of Smith & Eppinger (1997)

Smith & Eppinger (1997) aim to decrease process duration, an aspect of process behavior, by optimizing iterations within development processes by structural analysis. Therefore, they developed an extension of DSM to identify the controlling features of iteration. However, the assessment of the identified iterations stays on a qualitative level as well as does their proposed solution to decrease process duration with respect to iterations: faster iteration and/or fewer iterations.

Dynamic modeling of product development processes (Ford & Sterman, 1998a)

Ford & Sterman (1998a) developed an approach for the dynamic modeling of product development processes including the concept of rework due to iteration. They claim that to improve project performance it is necessary to understand the dynamic concurrence relationships that constrain the sequencing of tasks as well as the effects of and interaction with resources, project scope and targets. They use System Dynamics for their approach and propose a multiple-phase project model which explicitly models process, resources, scope, and targets. The model explicitly portrays iteration, distinct development activities and work constraints.

Interpretation of Ford & Sterman (1998a)

With their work, Ford & Sterman (1998a) added development processes to the domains being addressed with System Dynamics in the context of project management at this point of time (Browning et al., 2006). In agreement with Smith & Eppinger (1997) they also identified task relationships as a major challenge in product development. However, they have a different approach to the problem. By modeling the dynamic system behavior with System Dynamics they aim to significantly improve descriptions of development team mental models, of project constraints and of the drivers of project performance. They claim that the significant effects which process structures have on performance demonstrate the need for integrated project models. Their work, however, does not offer implications on how to improve the system structure.

A decision support tool for predicting the impact of development process improvements (Cronemyr et al., 2001)

Cronemyr et al. (2001) developed a decision support tool for predicting the impact of development process improvements. They extend the work of Smith & Eppinger (1997) by introducing the STAR (simulated to-be/as-is ratio) concept, which is a quantified measure of the impact of a suggested improvement. The authors use DSM as they consider it to be an easy tool to get a holistic view of a complex development process where flow charts become too complicated. However, they also discuss the correctness of the assumptions of Smith's and Eppinger's (1997) work transformation model, which they also use: linearity and stability of the process and fully parallel task execution.

Interpretation of Cronemyr et al. (2001)

With their work, Cronemyr et al. (2001) further developed the work transformation model approach of Smith & Eppinger (1997) by simulating process improvements and suggest a quantified measure to provide an estimate of the payback of the suggested change implementation. This can be seen as a bridge between the very detailed structural view of the analysis and the behavioral view of impact assessment of structural changes. Still, the simulation algorithm itself is rather simple, it makes various assumptions which may decrease the value of the analysis results, and it does not allow for modeling complex development systems with different kinds of involved domains, such as resources or fulfillment of project scope.

Complex Concurrent Engineering and the Design Structure Matrix Method (Yassine & Braha, 2003)

Within their work, Yassine & Braha (2003) propose a DSM-based unified modeling approach and solution to the four critical problems of concurrent engineering:

- iteration
- overlapping
- decomposition and integration
- convergence.

According to the authors, the convergence problem is concerned with the dynamic aspects of product development. They call the oscillatory development progress “Design Churn” phenomenon and claim that it is affected by the “Information Hiding” effect: A development team that has initially 100 tasks to work on will reduce the number of open problems with time until new information or feedback is received, in which case previously solved problems will either be reopened or new problems will be created. The new information or feedback is usually not given to the development team immediately, but received with some time delay (information hiding). Furthermore, the authors claim that the study of product development convergence and the “Design Churn” phenomenon cannot be treated as a single DSM due to time delays and asynchrony in information transfers. They refer to a more suitable dynamic DSM model, which has been developed by Yassine et al. (2003) (this is not the originally cited white paper, but the later journal publication), where the DSM notation has been expanded with numerical dependency strength, replacing the diagonal elements with the rate of work completion for each task. The DSM method in the context of concurrent engineering deals with the understanding and formulation of the interactions among many components of complex product development projects. According to the authors, the specific strengths of the structural representation as DSM are:

- Reducing the complexity of product development by providing a comprehensive analytical tool set.
- Easily and clearly revealing the information flows between the tasks of complex systems to visualize iteration.

Iterations which cannot be reduced by resequencing (with DSM) need to be strategically planned for in advance and incorporated in the project schedule. Furthermore, DSM (3) allows managers to discover previously unknown process, product and organizational patterns, which

opens new avenues for improvement and (4) suggests ways to mitigate the slow convergence or divergence of PD processes.

Interpretation of Yassine & Braha (2003)

Yassine & Braha (2003) claim that dynamic aspects of product development cannot be sufficiently treated with a single DSM due to the Design Churn phenomenon. This Design Churn phenomenon is similar to what Ford & Sterman (1998a) refer to as rework due to iteration. Yassine, Braha and other co-authors try to extend the DSM method for the Design Churn Phenomenon, but realize their dissatisfaction with the result (Yassine et al., 2003). However, they do not follow the approach of Ford & Sterman (1998a). DSM has its strengths in the understanding and formulation of the interactions in complex product development, but lacks the ability of sufficiently analyzing effects of iteration due to rework.

Optimal design process under uncertainty and reciprocal dependency (Suss & Thomson, 2012)

Suss & Thomson (2012) developed a model in which information flow is simulated explicitly such that the dynamic complexity of design processes with interdependent tasks is captured. Based on the observation that design processes are characterized by uncertainty and iteration, which makes it difficult to plan and manage them, the authors examined existing models that capture the dynamic complexity of engineering design projects with interdependent tasks. They claim that previous models are limited by their a priori assumptions about the amount of rework from empirical data in particular scenarios, linear theory or assumed probability distributions of rework. To solve these limitations Suss & Thomson (2012) consider the process a dynamic system with tasks, resources and information as its elements. They implemented a discrete event model, in which the modeled process can also be viewed as numerical DSM. The simulation model allows for a detailed examination of the effects of process structure, critical resource management, communication policies and uncertainty on the duration, effort and rework of the design process. After applying their model they find that simulation of complex product development processes can be helpful for planning and managing product development projects by developing and testing guidelines for improving information flow. This analysis could provide deeper understanding of mechanisms that drive performance.

Interpretation of Suss & Thomson (2012)

Suss & Thomson (2012) identified the need for a simulation approach to plan and manage design projects. They claim that existing approaches are limited by their a priori assumptions and developed a discrete event simulation considering tasks, resources and information as elements. The authors illustrate the modeled process within a DSM and thereby create a link between the static view on the process by the DSM and the dynamic view by the simulation model. Unfortunately, they do not elaborate on why they created the DSM view. As stated by Yassine & Braha (2003), this DSM representation of the process could also have been used for process analysis. Nor do the authors go into detail about why they chose discrete event as a simulation approach; but probably they chose it due to the detailed task level of their analysis. Due to the nature of their model, the authors identify a limitation of their model as it does not consider rework as a result of failure within tasks, even if the necessary information to conduct the task is available. This is a significant drawback, as both types of rework have to be

considered when analyzing engineering design processes: systematic (necessary in Engineering Design) and unsystematic (avoidable) rework.

Multi-level modeling and simulation of new product development processes (Karniel & Reich, 2013)

Within their work, Karniel & Reich (2013) present a multi-level modeling approach for supporting the management of product development processes. As a basis for their work Karniel & Reich (2013) differentiate between two types of process models: static level models such as DSM, which can be analyzed and simulated for estimating process completion time and other process parameters and dynamic level models, such as Petri nets in their case, which allow simulating the run-time of complex processes that include the parallel execution, iterations and other logic constructs. They draw the conclusion that neither DSM nor the dynamic modeling approach of Petri nets alone offer the complete modeling capabilities for complex development processes, and a multi-level approach must be adopted. They claim that both types of models are quite disconnected and there has been little attempt to integrate them into a multi-level model to support a more complete modeling of processes.

Especially, they note that in a previous survey (Karniel & Reich, 2009) of DSM-based plans in simulations a lack of clear translation from DSM representations to process schemes (defined as the process structure plus the progress logic) has been identified. Therefore they state that the integration of two different modeling approaches into a multi-level approach is not trivial.

Based on their study, they identify the need for modeling and simulating process transitions rather than relying only on either the static or the dynamic level without such transitions. They conclude that current methods not incorporating iterations and dynamic changes can provide only limited benefits for the planning of complex development processes.

Interpretation of Karniel & Reich (2013)

With their differentiation between static and dynamic level models the limitations of the previous models combining structural and dynamic complexity may be explained: These approaches were all extensions of either one of the two dimensions without bridging the complete gap to the other type of models. The multi-level approach of Karniel & Reich (2013) aims at bridging the complete gap between DSM and Petri nets by transforming one into the other. They state that the transformation itself is quite complex and that existing translation rules for DSM-based simulations are generally not sufficient. They illustrate their approach with a fictitious simple example and also note that additional aspects such as resource constraints or testing conditions need to be included in a multi-level modeling approach for product development processes. These aspects limit the usefulness of a DSM model for the translation, as only aspects of one domain can be modeled within it. Additionally, the transformation between the static and dynamic level models seems to be the critical issue for future work.

Conclusion

With respect to the differentiation of Karniel & Reich (2013) most previous publications of combinations of structural models and behavioral models in Engineering Design are extensions of either one of the two categories (static and dynamic level models) without bridging the

complete gap to the other type of models. The multi-level approach of Karniel & Reich (2013) tries bridging the complete gap between structural and behavioral models (for DSM and Petri nets) by transforming one into the other. They note that especially the transformation between the two levels is challenging. Rahmandad & Sterman (2008) further note that linking the behavior of a model to its structure becomes more difficult as model complexity grows. Additionally, a single DSM may not be the most suitable structural model for the deduction of process behavior, as only aspects of one domain can be modeled within a DSM. Behavior can be driven by various factors. Therefore this thesis proposes to use a Multiple-Domain Matrix as structural model.

For framework of this thesis System Dynamics is used as behavioral model. System Dynamics was defined by its developer John Forrester as “the study of information-feedback characteristics [...] to show how organizational structure, amplification (in policies), and time delays (in decisions and actions) interact to influence the success of the enterprise” (Forrester, 1958). This aligns well with the aims of this work. Even though other behavioral modeling approaches would probably be applicable as well (see Borshchev & Filippov (2004) for a review), System Dynamics is used due to its adaptable degree of abstraction, its management perspective and the distinct preliminary work within the System Dynamics community. Cooper (1980) defined the so-called rework cycle construct to model engineering design processes with System Dynamics. The construct was picked up by Ford & Sterman (1998a) and others and further developed.

It has to be mentioned that previous approaches of applying behavioral simulations to structural models exist for all three behavioral approaches mentioned in this thesis: Agent-based simulations (e.g. Lévardy & Browning (2009)), Discrete-event simulations (e.g. Meier et al. (2015)) and System Dynamics simulations (e.g. Storto et al. (2008), Laverghetta & Brown (1999), Hilmola et al. (2005), S. H. Lee et al. (2006), Lin et al. (2008)). Most existing System Dynamics models simulate the behavior without the basis of a structural model, whereas Agent-based and Discrete-event models are more likely to do so. This may be explained by the more bottom-up approach of these two models which seems to be more similar to structural modeling approaches than for System Dynamics with its top-down approach. The transfer of structural models such as DSM or MDM to Agent-based and Discrete-event models is more obvious, as the individual elements of e.g. a DSM model can easily be seen as individual agents and events. However, within the early phase of engineering design processes, process planning and management is often a more top-down than bottom-up approach as the elements and relations are not yet all known or defined. Thus, a successful combination of System Dynamics and Multiple-Domain Matrices is seen as most promising with respect to the objectives of this thesis.

3. Case studies

Within this section the iterative development of the framework for structure-based System Dynamics Analysis of Engineering Design Processes by case studies is shown. The beginning section addresses the structure of the case studies and provides an overview. Subsequently, the three most important and insightful case studies for the development of the framework for structure-based System Dynamics Analysis of Engineering Design Processes as well as the initial academic case are elaborated.

3.1 Structure of the case studies and overview

3.1.1 Structure of the case studies

Case studies contribute to theory building through observation of phenomena that have not been empirically tested. They are a way the make sure that researchers are making valid contributions and observations to the body of knowledge, especially in the context of lack of theory, complexity and the paucity of well-supported definitions and metrics. (Stuart et al., 2002)

Based on a literature review, Stuart et al. (2002) differentiates three types of case studies according to the stages of theory building:

- discovery/description/understanding
- mapping/relationship building
- theory validation/extension/refinement.

The herein presented case studies refer to the first two stages of theory building as the aim is to understand the correlation of structural and dynamic complexity of engineering design processes and try to deduce the behavior of engineering design processes from structural models. This can be understood as relationship building between static and dynamic views on engineering design processes.

All case studies follow the sequence of the stages for conducting the case by Stuart et al. (2002) (see Figure 3-1).

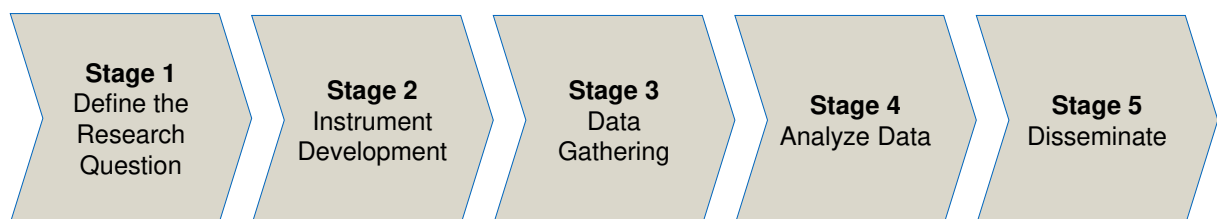


Figure 3-1 The five stage research process model (Stuart et al., 2002)

Stuart et al. (2002) describes the five stages for conducting the case as follows:

Stage 1: Define Research Question

This stage involves the definition of the research question. This includes contributing to the current body of knowledge and developing theory. Thereby, Stuart et al. (2002) cites Eisenhardt (1989) and Schmenner & Swink (1998) that good theory should be parsimonious, testable and logically coherent and result in new frame-breaking insights.

Stage 2: Instrument Development

After the definition of the research question, the measurement instruments to capture the data for future analysis need to be developed. The second step in conducting case research is the development of a research instrument and selection of the appropriate field sites. Stuart et al. (2002) specifically suggests the study protocol as instrument of choice. It encompasses the principal documentation that is needed to provide the necessary focus, organize the visits and ensure a thorough documentation. The instrumental development should have a focus but still keep some flexibility. Independently from the type of case study, it must demonstrate that its means of measuring are valid. The primary concerns in the context of case studies are construct validity (i.e., the quality of the measurements in reflecting the phenomenon) and internal validity (i.e., the quality of the measurements in reflecting the important relations inside the system). To ensure construct validity Stuart et al. (2002) cite Fielding & Fielding (1986), Herold et al. (2004) and Krause & Denzin (1989) for the concept of triangulation to use multiple sources of evidence for all important elements or variables of the case study.

Stage 3: Data Gathering

Stuart et al. (2002) refer to the term data as the recorded and transcribed information from the interview, documents that the object of observation is willing to provide and other observations of the researcher. Case study research faces a dilemma here: On the one hand the researcher must determine true causal relationships by building the interviewees trust quickly. On the other hand providing useful advice may become necessary for acceptance and trust which places the researcher in a conflict of interest. Therefore, it is important to reflect back to the protocol and the reason for the case researcher's presence.

Stage 4: Analyze Data

To analyze the observed, listened and recorded data experts refer to a need for lateral and conceptual thinking in the context of case study research. Important aspects are the abilities to extract significant patterns, simplify from descriptive information and to think laterally. A useful technique to see order from chaos involves structuring the data in a variety of patterns.

Stage 5: Disseminate

Stuart et al. (2002) state that case-based research may be subject to valid and invalid criticism. Therefore, it is important to clearly state why the research method of case study is appropriate to the research question at hand. He refers to Eisenhardt (1989) that case study research is especially appropriate in situations in which there is little previous literature or prior empirical evidence about a phenomenon and to Yin (1989) that the appropriate methodology depends on the current state of knowledge and the nature of the research problem.

3.1.2 Overview of the case studies

The framework for structure-based System Dynamics Analysis of Engineering Design Processes was iteratively developed by conducting case studies. This thesis includes the four major case studies for the development of the framework. Table 3-1 lists the case studies. For the first case study an academic setup was chosen to increase the observability of the engineering design process. Based on the insights from this first case study the framework was further developed and refined by conducting industrial case studies. The three most important and insightful case studies for the development of the framework for structure-based System Dynamics Analysis of Engineering Design Processes as well as the initial academic case are documented in the following sections. Each of the case studies was conducted in form of a student project in close collaboration with and supervision by the author. This setup ensured an application evaluation of the previously developed version of the framework on the special conditions of the next case study. Implications from the application of the current version of the framework were included in the next version of the framework.

Table 3-1 List of case studies documented within this thesis

No.	Name	Type
1	Engineering design process of the PSSycle	academic
2	Line process optimization at MAN Bus & Coach	industrial
3	Benchmarking process analysis at MAN Truck & Bus AG	industrial
4	CALA construction process	industrial

3.2 Case study 1: Engineering design process of the PSSycle

The case study was conducted in form of student projects (Produktentwicklung, 2013, 2014d) and the presented results base on the publications of Kasperek, Lichtenberg et al. (2015) and Kasperek, Lindinger et al. (2014).

3.2.1 Description of the case

The engineering design process examined in this first case study is the PSSycle project. The PSSycle project is part of the collaborative research centre “Sonderforschungsbereich 768 – Managing cycles in innovation processes”. Within the PSSycle project an innovative e-bike sharing system was developed. It was organized as a student project within the engineering department. A core team out of ten master students developed the e-bike sharing system within a time span of six months. While most of the students had a mechanical engineering background, some students had a computer science background. This was necessary due to the multi-disciplinary scope of the project. Additionally to the core team, other students examined various aspects of the project such as team composition or the engineering design process as such. (Kasperek & Maurer, 2013)

3.2.2 Stage 1: Research question

The research question addressed in this first case study can be stated as follows:

"RQ1: Can structural models of engineering design processes in form of Multiple-Domain Matrices, be used as a basis to simulate their dynamic behavior by System Dynamics models?"

If this research question would be approved this might offer the possibility of structure optimization towards a particular dynamic behavior of the engineering design process. This leads to a second subsequent research question for future research if RQ1 is approved:

"RQ2: Can existing structural analysis methods be extended based on the possibility of a structure-based System Dynamics Analysis?"

3.2.3 Stage 2: Instrument development

Throughout the design process weekly meetings were scheduled to discuss all relevant topics. These meetings were chosen as main source of information and therefore, all meetings were attended by the researchers and documented in detail. Based on the meetings, the current status of the engineering design process was documented every two to four weeks within a MDM. The changes within the MDMs over time give insights on changes within the structure of the engineering design process itself. A side effect of the regular attendances was building the development team's trust quickly, which is vital for the success of the case research as the case researcher must determine true causal relationships (Stuart et al., 2002). Additionally, all available documents such as requirements lists were used as input to track the current status of the engineering design process and to identify changes within the structure of the process as well as of the product to be developed. If the available data was not sufficient to verify the primary concerns of construct and internal validity, protocolled interviews with the affected students were chosen as additional instrument. This multi-faceted approach to gaining the complete picture was chosen based on the concept of triangulation.

3.2.4 Stage 3: Data gathering

Within this case study the gathered data was used to build a structural model of the engineering design process in form of a MDM. The structure of the current engineering design process was documented every two to four weeks with a MDM for each considered point of time. By investigating the change of the elements and relations incorporated in the particular MDMs over time, the change within the structure of the engineering design process itself over time was observed.

The domains of the MDM were chosen based on observations of the team meetings (the advantages of using a fixed metamodel such as the one of Kreimeyer (2009) were first identified in case study 3). The development team decided to break down their design problem into requirements first, then to have a solution neutral functional view on the system followed by dividing the problem into smaller sub problems for which solutions would be implemented. Finally the components should be assembled. The same structure was chosen for the MDM metamodel. Figure 3-2 shows the metamodel of the engineering design process.

Multiple Domain Matrix	Requirements	Functions	Hardware	Software	Service	Assembly
Requirements	influence	responsible for	define	define	define	
Functions		influence	influence	influence	influence	
Hardware			may change	influence		influence
Software			influence	may change	influence	influence
Service				influence	may change	influence
Assembly						influence

Figure 3-2 Metamodel of the engineering design process (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

Each domain can be seen as one phase of the engineering design process:

- Requirements domain: phase of defining requirements
- Functions domain: phase of defining functions to fulfil the requirements
- Hardware domain: phase of implementing hardware components
- Software domain: phase of implementing software components
- Service domain: phase of implementing service components
- Assembly domain: phase of integrating the sub components.

The process can only be partly understood as sequential: While the phases Hardware (HW), Software (SW) and Service (S) were conducted in parallel, there were also iterations within and between the other phases.

Interdependencies were only chosen, if the connection between the particular domains was important to describe the structure of the engineering design process. The different types of interdependencies between the domains that were modeled within the MDM are:

- Requirements are responsible for Functions.
- Requirements define the HW, SW, S components.
- Functions influence the HW, SW, S components.
- HW, SW, S components influence the Assembly.
- HW and S components influence the SW components.
- All elements of the domains are able to influence elements of the same domain.

As the development team started with the requirements definition and ended with the assembly, the metamodel was detailed with elements and relations throughout the development process "from left to right". The necessary information was gained from available documents such as the requirement list, through workshops with the developers as well as through detailed records of each team meeting. Especially within the early phase of the process the elements and

relations considered by the development team and documented by the case researcher changed frequently. With ongoing completion the amount of changes decreased.

3.2.5 Stage 4: Analyze Data

The evolution of the engineering design process structure over the time of the project was used as a dataset to build a System Dynamics model of the structure of the engineering design process. Therefore, as a first step, the requirements of the System Dynamics methodology are mapped on the special conditions of structural models. This led to the development of a qualitative System Dynamics model of the structural data. This model was used to illustrate and relate the various dynamic influences on the structure of the engineering design process. The qualitative model was further used to develop a quantitative model to enable a simulation of the engineering design process based on its underlying structure and thereby to refine the model as well as the model transformation method. To model the qualitative System Dynamics model the software tool Vensim® was used.

Qualitative System Dynamics modeling

Building a System Dynamics model out of an MDM is a process that can be divided into the consecutive development of a qualitative and a quantitative System Dynamics model. As a first step the six DSMs reflecting the relations between the elements of the particular domains were transformed into rework cycles which represent the major concept in System Dynamics to model iterations (Ford & Sterman, 1998a). The same qualitative rework cycle concept was used for each rework cycle. Figure 3-3 illustrates a detailed view on a particular rework cycle.

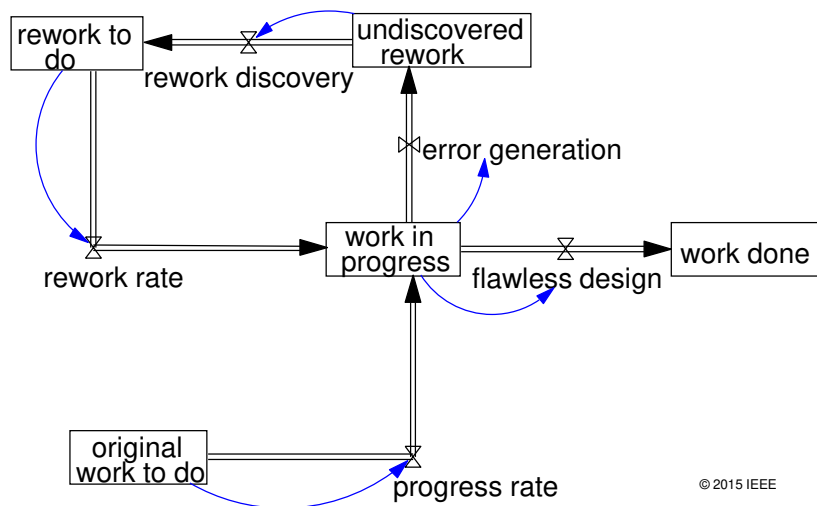


Figure 3-3 Detailed view of a single rework cycle (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

The rework cycle operates as follows: At the beginning all tasks are stored at the stock *original work to do*. The tasks then flow with a certain *Progress rate* to the stock *work in progress*. Ideally the tasks are completed in this stock and if the design is flawless the tasks will reach the *work done* stock. Unfortunately, not all tasks are completed flawlessly as errors occur.

Therefore as a first step these errors lead to *undiscovered rework*. Only if these errors are detected, rework will be necessary. The number of tasks that have to be reworked is described by the *rework discovery* which processes the tasks to the *Rework to do* stock. The actual amount of tasks that are reworked is processed by the *Rework rate* which processes the tasks from the *Rework to do* stock back to the *work in progress* stock. (Kasperek & Maurer, 2013; Lyneis & Ford, 2007)

However, rework can cause more rework as errors in design are often detected some time after their occurrence (Lyneis & Ford, 2007).

The interdependencies between the domains / phases were modeled as connections between the rework cycles. Each connection between two rework cycles had the following components:

- process concurrence: Task B is only active if Task A is finished to a certain percentage
- feedback of rework: While task B is active not only rework within this task itself is discovered but also necessary rework within task A. Thus the Rework discovery rate of task A is triggered by task B.

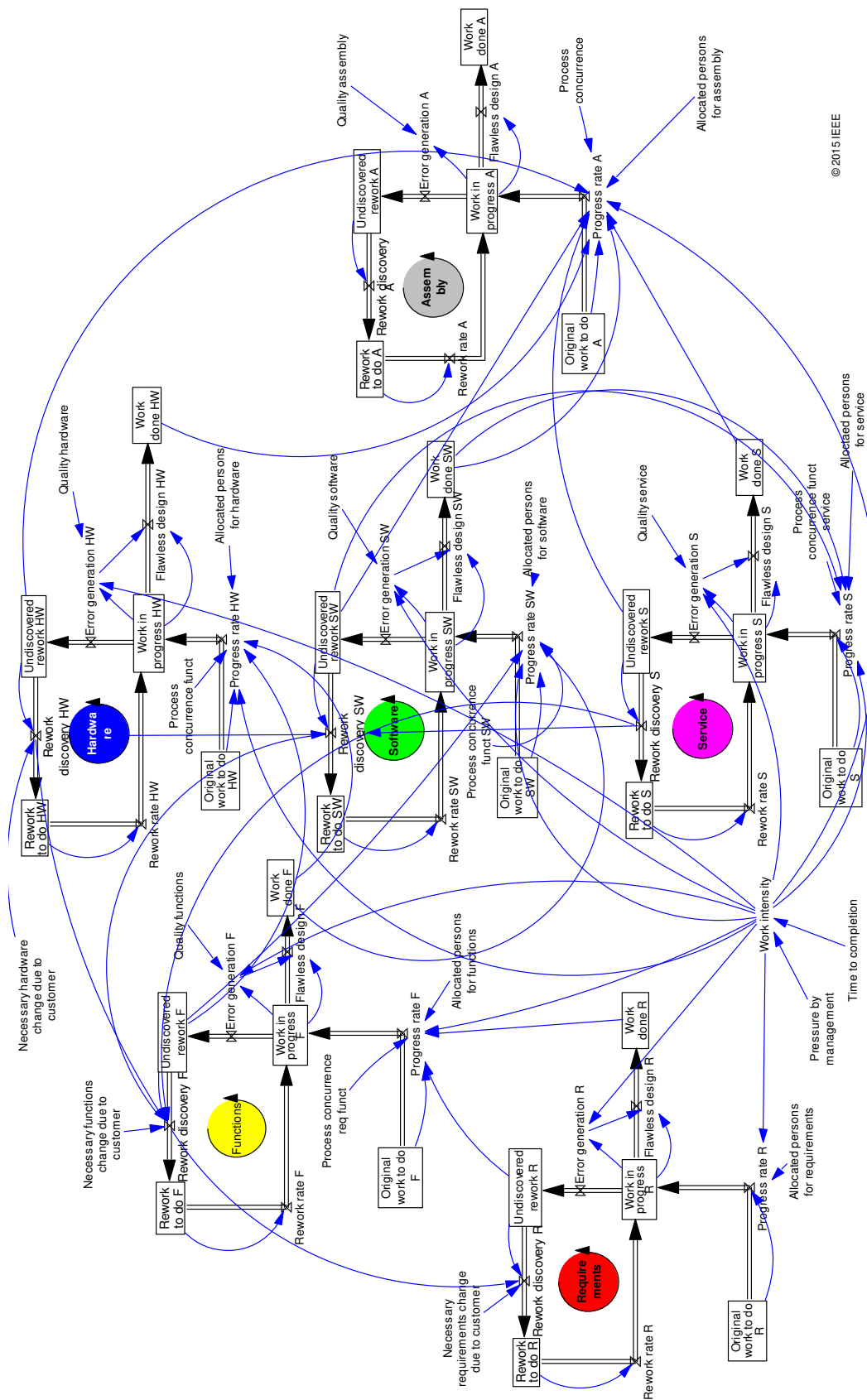
These connections were modeled as follows:

The interdependencies *influence* and *responsible for* are transformed into vectored connections between the stocks *Work done_X* to *Progress rate_Y* for the feedforward case and into vectored connections from *Rework discovery_Y* to *Rework discovery_X* for the feedback case. *X* and *Y* are thereby indices for the involved domains. Together with the additional variable *Process concurrence_XY* this allows to model various interaction scenarios: While through the connection *Work done_X* to *Progress rate_Y* the performance within phase B can be controlled based on the status of *X*, *Process concurrence_XY* can trigger the optimal amount of concurrence between *X* and *Y*. The connection from *Rework discovery_Y* to *Rework discovery_X* on the other hand can be directly interpreted as iteration within a previous phase triggered by a subsequent phase. For example: Within the implementation of a HW component it is noticed, that the defined function for the component is not correct. Thus, this definition has to be refined before the HW component can be further detailed.

The interdependencies *define* between *Requirements* and *HW*, *SW*, *S* were not implemented within the qualitative System Dynamics model as these were already implemented indirectly by the dependencies: *R is responsible for F* and *F influences HW, SW, S*.

As suggested by Ford & Sterman (1998b), the variables *Work intensity*, *time to detect errors*, *Pressure by management* and *Quality function* were implemented within the model. Master students were allocated by the variable *Allocated persons for X*.

The derived qualitative System Dynamics model is illustrated in Figure 3-4. As For arrangement reasons the indexes of the rework cycles are abbreviated (R [red], F [yellow], HW [blue], SW [green], S [pink], AS [grey]).



© 2015 IEEE

Figure 3-4 Qualitative System Dynamics model of the PSSycle development process (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

Quantitative System Dynamics modeling

The values for the quantification of the model were defined based on the input from the measured process data and from interviews where the measured data was not sufficient. Comparing the simulated values with the measured values showed a good approximation of the real process. Depending on the characteristics of the particular process step, different values respectively equations were implemented for each rework cycle. Figure 3-5 shows the exemplary simulation results of the requirements rework cycle. The *Original work to do* decreases continuously until it reaches 0 while the other stocks increase. At the end of the simulation all work packages have moved from *Original work to do* to *Work done*. The *Work done* stock does increase continuously to the same amount as *Original work to do* decreases. This is due to occurring rework: Work packages still reach the *Work done* state even though all *Original work to do* is already 0 due to work packages that had to be reworked. The *Undiscovered rework* reaches its maximum at the point in time when all work has been conducted once.

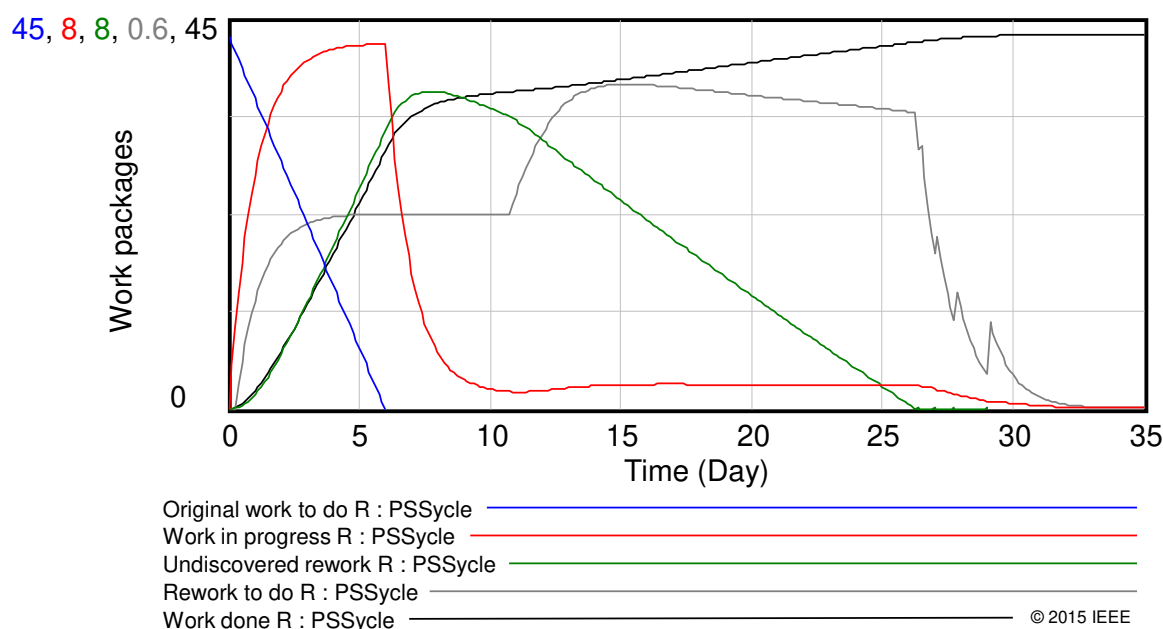


Figure 3-5 Exemplary simulation results of the requirements rework cycle (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

Overall 463 work packages were identified for the six domains / phases. The amount of *Original work to do* for each rework cycle was originally based on the amount of elements of particular domain within the MDM. This was changed as initial considerations showed that the amount of elements did not correlate with the amount of work to be conducted within each task. Instead of that capability planning, time lines and other available documentation was used to estimate the work to be done within each domain / phase. The work to be done is reflected by the amount of elements of each domain. Table 3-2 shows the estimated original amount of work packages as well as the amount of elements of the particular domains within the MDM.

Table 3-2 Estimated amount of work packages and elements within the MDM of each domain (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

Domain = rework cycle	Amount of elements	Amount work packages
Requirements (R)	43	43
Functions (F)	20	40
Hardware (HW)	17	180
Software (SW)	5	140
Service (S)	3	40
Assembly (A)	10	20
		Σ 463

The 463 work packages were executed by ten master students. The master students and the work packages were directly allocated to the corresponding domains. The requirements (R) and functions (F) work packages were subsequently conducted by all ten master students. The corresponding *Process concurrence rate* was set to 0. The development of hardware components (HW) was conducted by five master students and started before the F work packages were completely finished. The *Process concurrence rate* was therefore set to 0.1. Compared to HW, the development of software started earlier: As soon as it was obvious that some functions would probably be implemented by software (SW). The *Process concurrence rate* between F and SW work packages was therefore set to 0.5. The services (S) were implemented by one master student which started at the same time than the SW team. The *Process concurrence rate* was therefore also set to 95%. The assembly was started as soon as the bulk of HW and SW components were developed. The *Process concurrence rate* was therefore set to 0.1.

One of the main modeling assumptions was that one master student is able to conduct one work package at one day on a long time perspective without causing above-average rework. Assuming optimal efficiency and no rework the 463 work packages could be theoretically processed within 47 days of work (10 master students, 1 work package per master student per day). The real process including public holidays and the exam period of the master students took 105 days. These periods were also incorporated within the simulation model as the *Time to completion* influenced the modelled *Work intensity*. The *Work intensity* is also influenced by the *Pressure by management*. If the *Pressure by management* increases the *Work intensity* also increases. On the other hand the *Error generation rate* and therefore the amount of created rework is dependent on the *Work intensity*. The *Work intensity* was set to 1 at the beginning. A value of *Work intensity* lower or equal to 1 led to 20% rework while more rework (30%) was created in periods of higher *Work intensity* (>1). Figure 3-6 illustrates the effects of work intensity and pressure by management on the error generation during the HW phase. The work on the HW work packages started on day 59, therefore no errors were generated before.

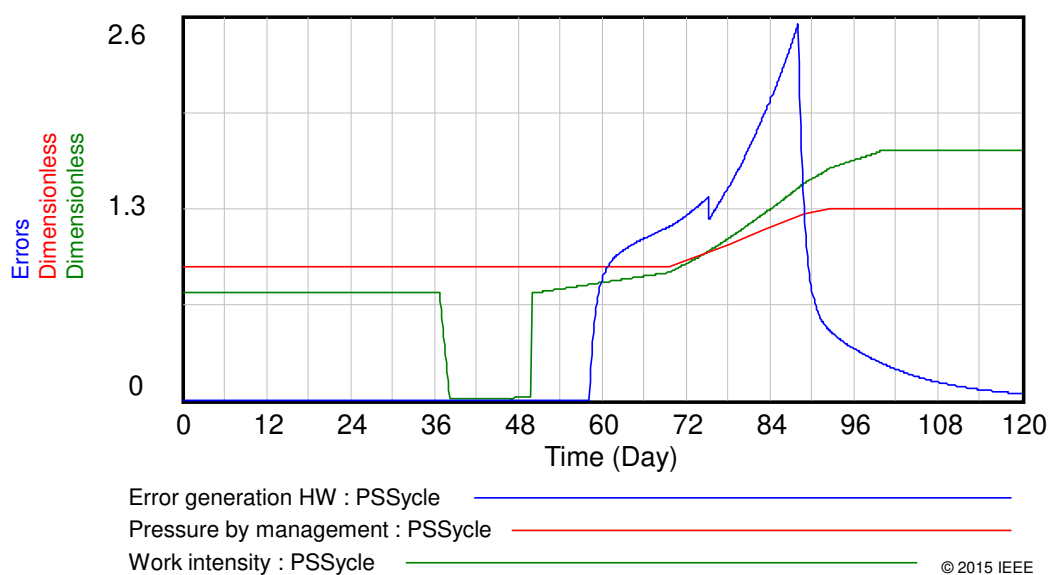


Figure 3-6 Exemplary effect of work intensity and pressure by management on error generation HW (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

Simulation of the engineering design process

Figure 3-7 shows the values of all *Original work to do* stocks over the simulation time. The *Original work to do* of the functions work packages decreases first, then stops and decreases further. This is due to the influence of *Process concurrence* to the previous process. During the conduction of the functions work packages necessary rework on the requirements work packages is discovered which triggers this rework cycles again and stops the functions rework cycle due to the *Process concurrence* condition. The same effect can also be seen for assembly work packages. The constant period of the SW and S graph between approx. day 38 and day 50 can be retraced to the *Time to completion* variable reflecting the public holiday and exam period. The HW rework cycle starts quite late which can be explained by the low degree of *Process concurrence* with the functions rework cycle. This on the other hand triggers the degree of completion of the A task due to the high degree of dependence between these processes.

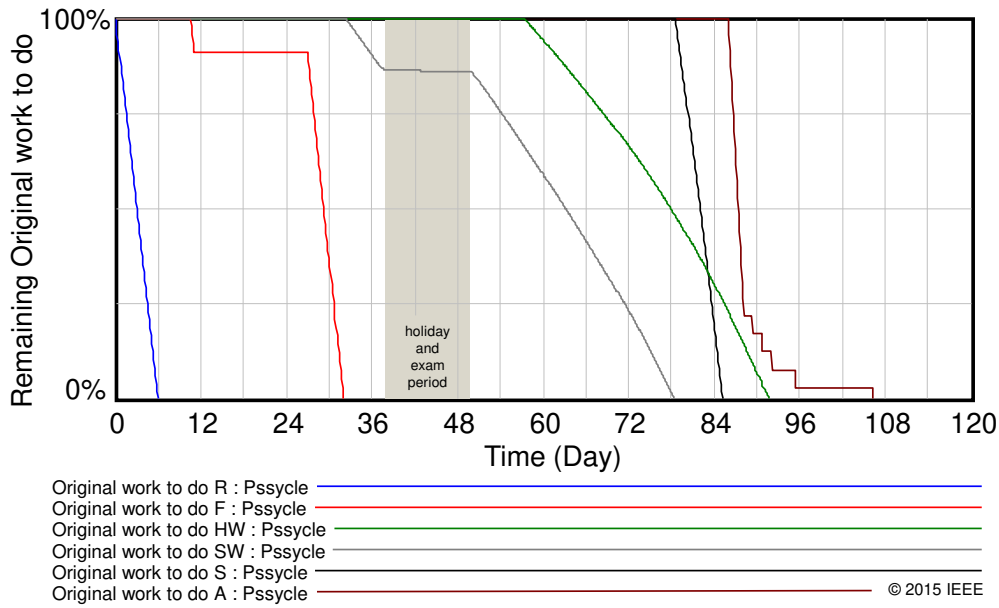


Figure 3-7 Simulated Original work to do stocks over time (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

Additionally to the previously described aspects, Figure 3-8 illustrates the influence of rework of work packages on the overall *Work done* of the phases. The first increasing phase of each graph of Figure 3-8 can be assigned to the primary processing of the work packages. The increase after the inflection of the graphs can be explained by rework. As rework is not discovered immediately the *Rework discovery rate* and *Rework rate* increasing the *Work done* stock are lower than the original *Progress rate*.

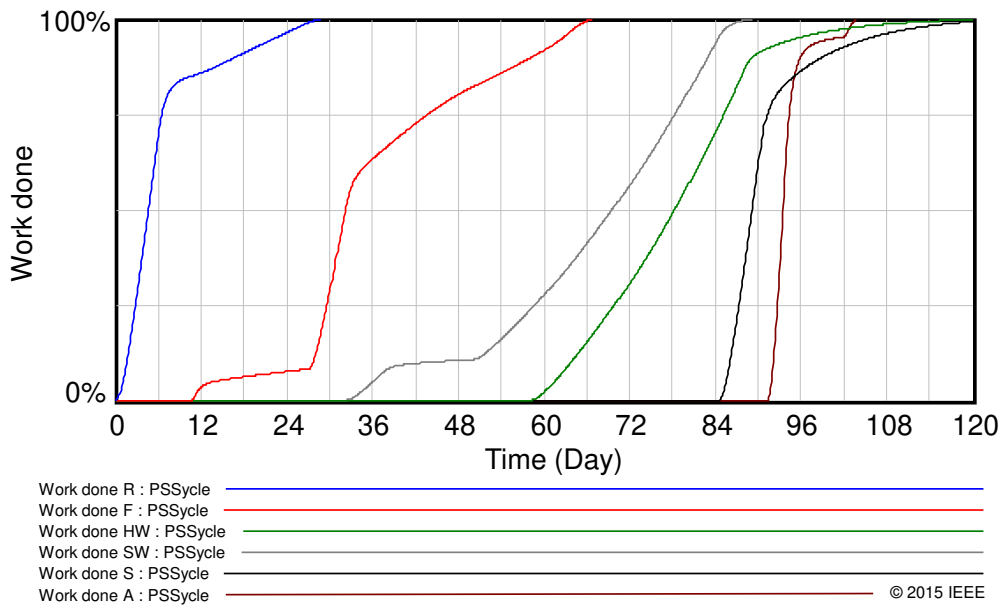


Figure 3-8 Simulated Work done stocks over time (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

Figure 3-9 shows the influences of the particular *Rework discovery rates* on each other. *Rework discovery* in subsequent process tasks also increases the *Rework discovery* in previous tasks as the reasons for errors might have occurred within earlier tasks. The graphs of the particular *Rework discovery rates* show this coupling, for example the increasing *Rework discovery F* after 27 days erratically re-triggers *Rework discovery R* again. Another example is the *Rework discovery S* which erratically reactivates *Rework discovery SW*.

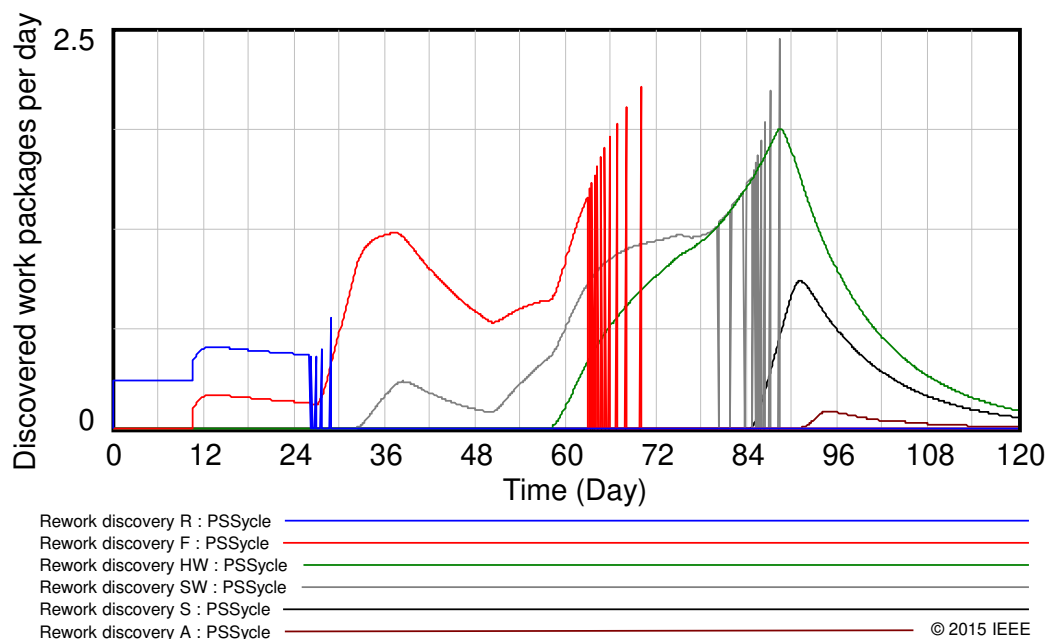


Figure 3-9 Simulated Rework discovery rates (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

The System Dynamics model simulates the duration of the overall engineering design process. Neglecting minor rework on the service work packages, the simulated engineering design process of the PSSycle took approx. 106 days.

3.2.6 Stage 5: Disseminate

The System Dynamics simulation allows investigating the run-down of the engineering design process and thereby serves as a method for the:

- generation of basic understanding of the engineering design process dynamics
- decision support through simulation experiments
- process optimization.

The following insights considering the project duration were drawn based on the simulation model:

While the PSSycle process was originally planned to be finished after 100 days, the overall simulated process duration of 106 days is very close to the real duration of 105 days. The duration of the particular phases could be also simulated very close to reality. In both cases, the

simulation and reality, the process delay could be retraced to the public holiday period within the period of the engineering design process.

Figure 3-7 shows that especially the Hardware phase increased the overall project duration. Due to the low degree of possible Process concurrence with the previous Functions phase, the work on Hardware started quite late. This also caused additional rework at the last Assembly phase. Based on the simulation two optimization potentials were identified: Reducing the error rate, thus the amount of rework within the Function phase, or accelerating the Hardware phase.

The error rate of the Functions work packages could have been decreased by an intentional actuating of the phase by the management. Another possibility would have been sensitizing the master students of the high error rate of their work packages.

The Hardware task could have been accelerated by reallocating the assigned master students. If one master student, is reassigned from the Software phase to the Hardware phase, the work done of the Assembly phase increases more quickly. However, the complete Assembly work would be done later (due to rework effects). Figure 3-10 illustrates this effect graphically.

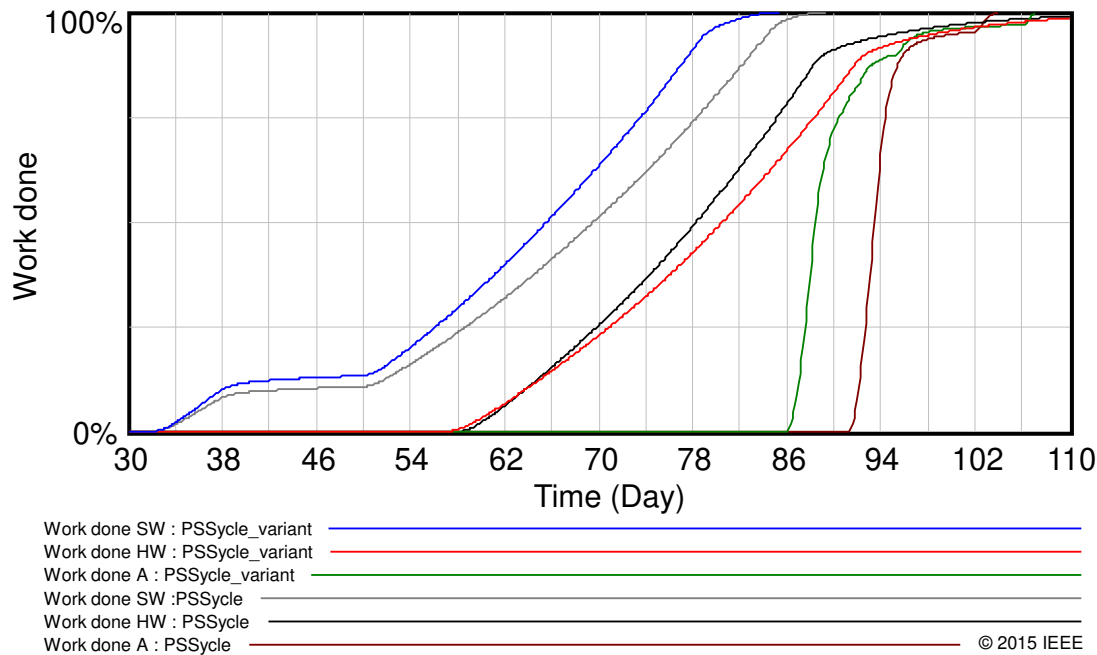


Figure 3-10 Original simulation compared to variant with one student reallocated from SW to HW work packages (Kasperek, Lichtenberg, Maisenbacher, Hollauer, et al., 2015)

The simulation also allows estimating the influence of quality management on the overall process. If the amount of erroneous work packages that are allowed is increased from 0% to 5% the simulated process time decreases by 25%. The same effect can be also observed in the other direction: 5% more rework causes an increase of development time by 25%.

3.2.7 Implications of the case study

In this case study the domains of the MDM represented the different phases of the engineering design process. These domains were transformed to rework cycles, while the information incorporated by the particular elements and their relations was aggregated. Initially it was considered to transform the particular elements of the domains into rework cycles as these represented the actual work packages. As this would have led to more than 400 rework cycles and also a reasonable clustering of elements was not possible due to the different amount of work reflected by the elements, the transformation was conducted on the domain level.

General steps of the structure-based System Dynamics Analysis framework concept

Based on the insights from this first case study the general steps for the desired framework concept were identified. Figure 3-11 illustrates the four general steps of the framework:

1. System structure: Within this step the system structure has to be defined and modeled by a MDM.
2. Transformation: Within this step the system structure representation as MDM is transformed into a System Dynamics model which enables the analysis of the dynamic behavior of the engineering design process.
3. Dynamic behavior: Within this step the behavior of the system based on its structure is analyzed by simulation of the System Dynamics model.
4. System Structure: Within this step the insights from the analysis of the dynamic behavior are transferred back to implications on the system structure.

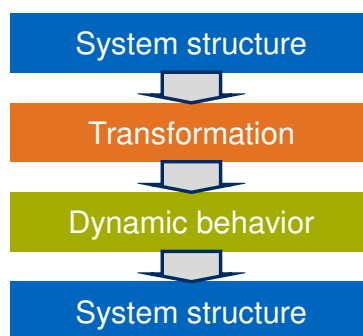


Figure 3-11 Core elements of the framework concept

Model transformation concept from MDM models to System Dynamics models

Another implication from the case study was that building a System Dynamics model out of an MDM is a process that can be divided into the development of a qualitative and a quantitative System Dynamics model. The skeleton of the qualitative System Dynamics model can be transformed from the MDM model.

Domains of the MDM can be transformed into one System Dynamics rework cycle per domain. It has to be mentioned that domains may not be necessarily transformed into rework cycles: An early version of the MDM also included the domain *Status*, which described the degree of

maturity for each function and component. However, there was no need to transform it into the System Dynamics model.

Interdependencies between the domains can be transformed into relationships between rework cycles.

Use of multiple rework cycles within one model

Furthermore, rework cycles were identified as suitable to model the typical iterations of engineering design processes. However within literature usually one single rework cycle is used to model the complete process. If the inherent sequence of phases, or the allocation of resources to phases is changed, these changes cannot be easily incorporated in the System Dynamics model. Therefore, the assessment of different process sequences of engineering design processes remains a challenge.

To solve this challenge multiple rework cycles were used in the presented System Dynamics model.

Structured representation of dependencies with the System Dynamics model to reduce complicatedness

The large amount of relations that had to be modeled manually in the System Dynamics model with its rework cycles led to a quite complicated model. With the growing model it got more and more difficult to keep track of the incorporated relations within the model. However, the analyzed process was small and had a low complicatedness compared to large industrial engineering design processes. Therefore, a more systematic way to cope with the amount of relations within the model should be identified.

3.3 Case study 2: Line process optimization at MAN Bus & Coach

Based on the insights and implications from the first case study with an academic example, case study 2 was applied within an industrial context. It was conducted in form of a student project (Produktentwicklung, 2014a) and the presented results base on the publication of Kasperek, Bermond et al. (2015).

3.3.1 Description of the case

Process optimization is an ongoing challenge and the need for change increases constantly as business environments are becoming more and more competitive (Hale, 1997). This case study was conducted in collaboration with MAN Bus & Coach (Pty) Ltd. South Africa (MBC). The industry partner identified the need for a process change: For the considered factory, an increasing amount of produced variants caused an increasing number of arriving parts. The so-called “warehouse team” was unable to cope with the increased variety of parts. The location of the site was in a newly industrialized country (South Africa) and the team consisted mainly of non-specialized workers, so the risk of process errors during changes on the current allocation of persons to line process steps was high. To reduce this risk, the need for a process simulation as a safe testing environment was identified. To create a holistic process simulation,

a structural model in form of a MDM provides the basis to develop a System Dynamics model. The System Dynamics model, in turn, provides a simulation environment for the purpose of analysis and optimization.

In contrast to the first case study, here a business process in the special form of a line process is observed. In contrast to engineering design processes, business processes are repeatable without the necessity to generate knowledge about the process execution (Kreimeyer, 2009, p. 64). This repeatability of the process offers the possibility to assess process changes as the outcomes of the process can be measured before and after the change. In this case study the outcomes of the line process were measured before and after a structural change which is usually difficult for engineering design processes due to the low repeatability. The structural change was chosen based on the structure-based System Dynamics analysis of the process. Thus, the result of the deduced structural change was measured.

3.3.2 Stage 1: Research question

From the perspective of the industry partner the question of the case study can be stated as: “How can workers of the line process be optimally allocated to the process steps?”

This allowed addressing the general question of interest from the research perspective: “Can the concept of structure-based System Dynamics Analysis, which is developed for engineering design processes, also be applied to line processes?”

If so: “Can additional implications for the structure-based System Dynamics Analysis framework be identified?”

3.3.3 Stage 2: Instrument development

The general steps of the process could be extracted from existing flow charts. Additionally, protocolled interviews with the affected workers and managers were chosen to develop a general overview of the process and to derive the allocation of workers to the process steps. Various workshops with the warehouse manager iteratively refined the data and verified the primary concerns of the construct validity and internal validity. Due to the increasing knowledge of the researchers during the period of the case study, the MDM models were refined over time several times. If possible, directly acquired information within the MDM was reviewed by calculation the same subsets also indirectly and comparing and scrutinizing the results.

3.3.4 Stage 3: Data gathering

The focus of the data gathering was the line process. Figure 3-12 shows a simplified representation of the examined line process. Information about the process was derived from the company’s process charts and from interviews with the management team. Each part arriving triggers the process.

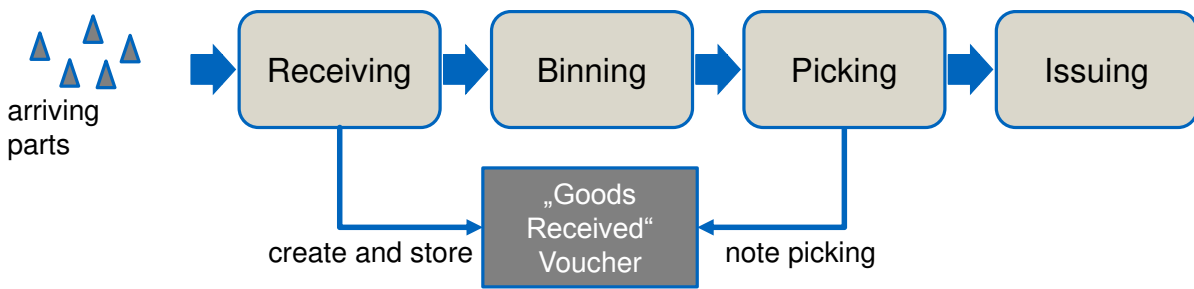


Figure 3-12 Simplified line process

The presented line process at the industry partner took place in challenging environmental conditions. Additionally to weather influences on the local infrastructure, language difficulties were another factor that caused complexity. A variety of official languages and dialects existed in the country where the factory was located. The education level varied between the employees. Some were not able to write properly and while they were able to speak English, most preferred to communicate in their native languages. This often resulted in misunderstandings.

The gathered information was used to build a MDM of the process. Process steps, persons, resources, and effort were chosen as domains. Figure 3-13 shows the MDM metamodel for the line process.

Multiple Domain Matrix	Process step	Persons	Resource	Effort	
Process step	is followed by			has	
Persons	amount allocated to		$(P-PS)*(R-PS)^T$		
Resource	required for				
Effort			founded in using		

Legend:
 P-PS Persons - Process step subset
 R-PS Resource - Process step subset
 ()^T Transposed

© 2015 IEEE

Figure 3-13 Metamodel of the line process MDM (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

For domains where the particular elements could not be identified by the process charts, additional input was given by the responsible warehouse manager. He also provided the current allocation of personnel at the warehouse. The effort-domain ranked work-intense process steps.

The considered interdependencies were: Process step is followed by process step, process step has effort, amount of persons is allocated to process step, resources is required for process step and effort is founded in using resource. The interdependencies between persons and resources were deducted indirectly by calculation. The domain effort can be seen as a scale, containing “high”, “medium” and “low” measures, indicating the effort required to complete a process step. Figure 3-14 shows the MDM of the line process.

Multiple-Domain Matrix		Process st.				Persons				Resource			Effort		
		Receiv.	Binning	Picking	Issuing	Receiver	Picker	Team l.	Line r.	Wareh.	Embrace	BoM	Low	Medium	High
Process steps	Receiving	0	1	0	0								0	0	1
	Binning	0	0	1	0								1	0	0
	Picking	0	1	0	1								0	1	0
	Issuing	1	0	0	0								1	0	0
Persons	Receiver	5	0	0	0					1	1	0			
	Picker	0	2	2	0					1	0	0			
	Team leader	0	0	0	3					1	1	1			
	Line runner	0	0	0	5					0	1	1			
Resource	Warehouse	1	1	0	0										
	Embrace	1	0	1	1										
	BoM	0	0	0	1										
Effort	Low									0	0	0			
	Medium									0	1	1			
	High									1	0	0			

© 2015 IEEE

Figure 3-14 MDM of the line process (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

3.3.5 Stage 4: Analyze Data

Model requirements

For this case study, the transformation method of case study 1 was used as a basis. Additionally, requirements for the desired capabilities of the System Dynamics model were defined to assess if the System Dynamics model sufficiently enough represents reality for the intended purpose of the case study. The requirements are loosely based on the verification criteria for System Dynamics models suggested by Sterman (2000). Table 3-3 lists the requirements for the System Dynamics model.

Table 3-3 Requirements for the System Dynamics model adapted from Kasperek, Bermond, et al. (2015)

No.	Requirements for the System Dynamics model
1	The process steps shall be conducted in the correct order.
2	The allocation of persons to process steps shall have a measurable influence.
3	The change of the amount of persons at a single process step shall have a measurable impact on completed work packages.
4	The increase of the amount of persons shall induce an increase of overall process performance.
5	The decrease of the amount of persons shall induce a decrease of overall process performance.
6	Changing the process sequence shall impact the overall process performance.
7	External impact factors shall influence the overall process performance

Qualitative System Dynamics model

Based on the gathered data, the qualitative System Dynamics model could be built. The process steps receiving, binning, picking and issuing were transformed into one rework cycle each. A resource pool represents the persons domain. The values of the resources and effort domains were used to quantify the model and therefore are not directly represented by the qualitative model. Manual workforce is usually not free of human error and thus, unplanned iterations can always occur to a certain extent: One finding of this case study is that every element of the MDM (e.g. a process step) that contains manual workforce should be considered to be modeled with a rework cycle, while elements that do not contain manual workforce may be modeled in different ways more easily or should be neglected.

For each of the four process steps a four stock rework cycle concept based on Lyneis & Ford (2007) was chosen for the System Dynamics model, see Figure 3-15.

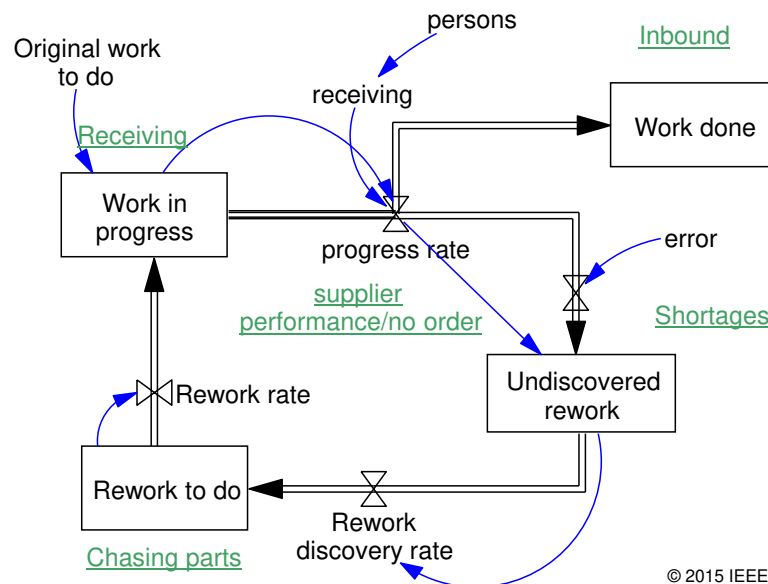


Figure 3-15 Rework cycle concept (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

The rework cycle slightly differs from the one used in the first case study as the *Original Work to Do* is modeled as variable and not as stock and the separation of *Work in progress* into *Work done* and *Undiscovered rework* is modeled differently. Furthermore, the System Dynamics model differs significantly from the first case study in how the rework cycles are connected: In the first case study the process concurrence of the rework cycles was based on the percentage of work done within the predecessor rework cycles (equivalent to process phases), and each rework cycle (process phase) had its own amount of original work to do. In the second case study, the original work to do from rework cycle 1 “flows” through the rework cycle to rework cycle 2. This modeling concept was chosen due to the characteristics of a line process: The parts that are received are all binned afterwards. Equivalently all parts that are picked will be issued afterwards and thus, flow from rework cycle 3 to rework cycle 4. As the parts are stored in a shelf between binning and picking and not all parts binned during the simulation period may also be picked, rework cycle 2 and 3 are not directly connected via a flow.

Figure 3-16 shows the qualitative System Dynamics model.

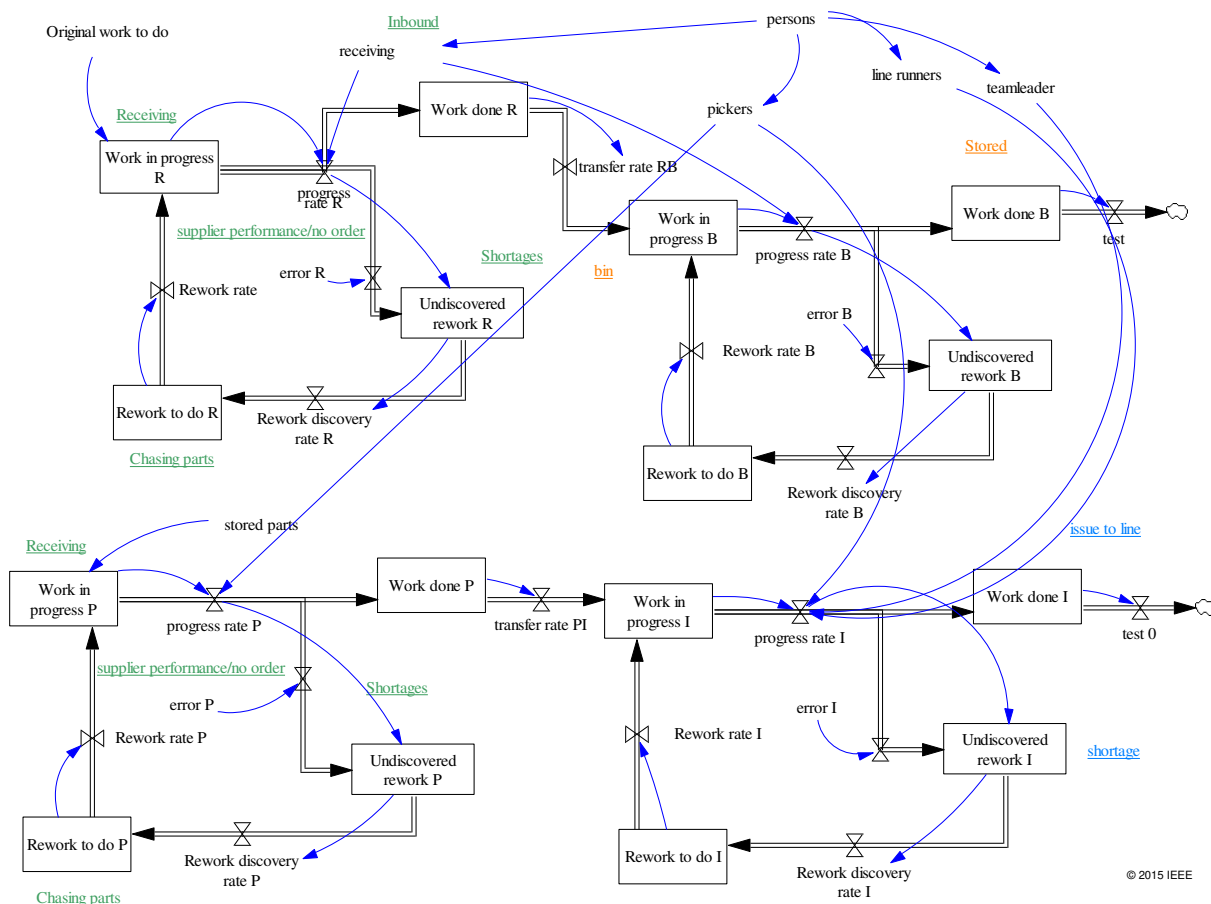


Figure 3-16 Complete System Dynamics model of the case study (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

The four used rework cycles were named according their originating element of the process step domain of the MDM, *R* for receiving, *B* for binning, *P* for picking, *I* for issuing. The first rework cycle represents the process step of receiving. The parts to be processed are generated by the *Original work to do R* variable and flow through the rework cycle in the *work done R* stock flow with a delay into the *Work in progress B* stock, which is the beginning of the second rework cycle representing the process step of binning. After processing the parts through this rework cycle the binning process step is finished and the parts are stored in a shelf. As described above the rework cycles of binning and picking are not connected by a direct flow. Instead, the variable *Stored parts* was implemented, which works similarly to the *Original work to do* variable.

The blue arrows in Figure 3-16 originating from the *persons* pool indicate the interdependencies, which originates from the *Persons allocated to Process Steps* subset of the MDM.

Quantitative System Dynamics model

The model was quantified based on the data acquired during the modeling of the MDM. Further input from measured process data and interviews was acquired where the measured data was not sufficient. For the measured error rates of the process, it was not possible to find any correlation function. Therefore the error rate was implemented by a random function. The measured maximum and minimum error values over time were used as limits for the random function. Comparing the simulated results with the measured process data showed a good approximation of the real process.

Testing of System Dynamics model

The model was tested based on the initial requirements for the simulation indicated in Table 3-3. The initial allocation of persons to the particular process steps (6 pickers, 3 team leaders, 3 receivers, 5 line runners) was varied in different scenarios. The resulting simulation runs were then discussed with and assessed by on-site process. For the purpose of this case study the focus of the test was verifying that the System Dynamics model was able to express the expected dynamics of the process qualitatively. The correctness of the simulated extent of movement of the considered variables was not tested.

Figure 3-17 shows the simulation results which were used for the testing. The red graph indicates the values of the simulated *Work done I* stock, thus, showing the output of parts issued to the production line. The green graph indicates the output of parts issued if the total number of workers is increased from originally 17 to 20. As expected the output of parts issued increases faster than the initial situation's graph. When the number of total workers is reduced to 14, the graph should increase with a flatter angle than the initial situation's graph, this simulation showed the same behavior as indicated by the blue graph.

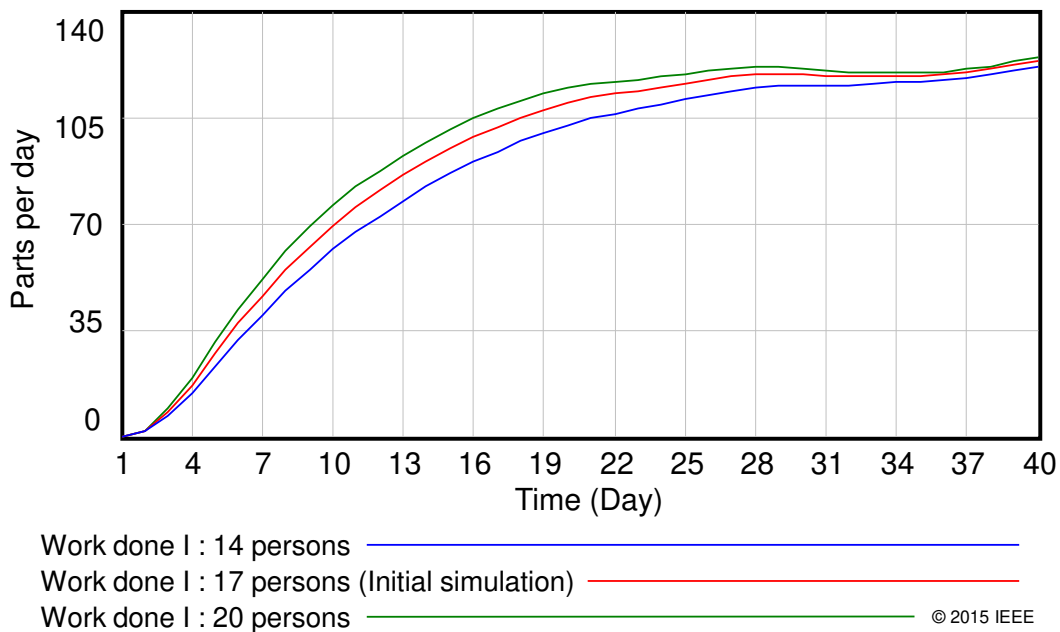


Figure 3-17 Comparison of "Issuing work done" for 14, 17 and 20 workers (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

In the next step, the sensitivity of the model with respect to the amount of occurring errors was examined. If the simulation approximates reality correctly, a reduction of the error rate should increase the output of parts while an increase should decrease the amount of parts. Figure 3-18 shows the behavior of the simulation model which show the expected real behavior.

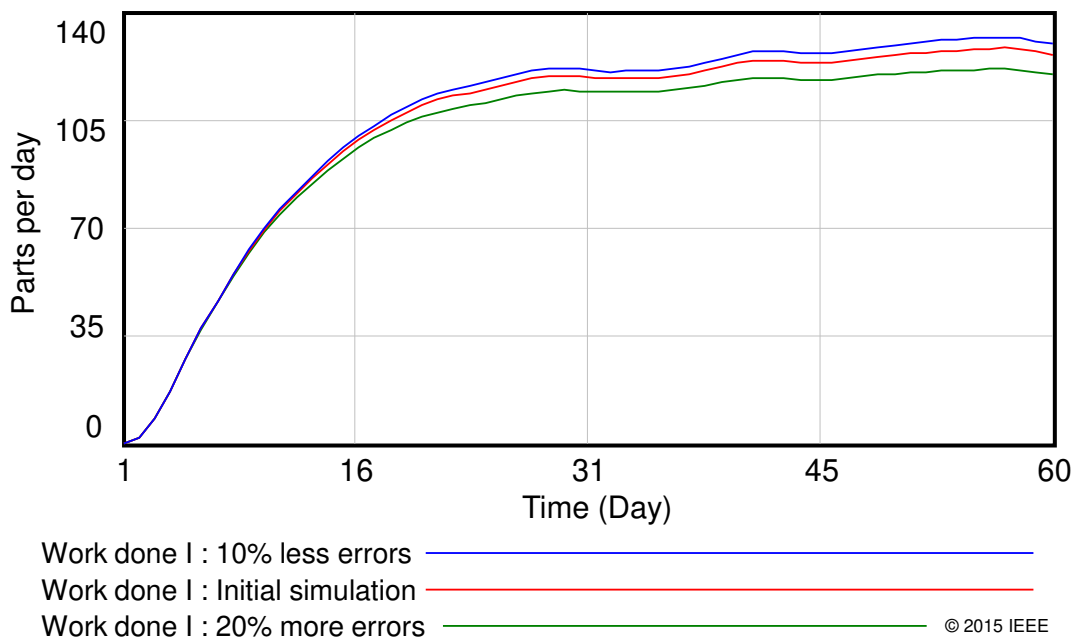


Figure 3-18 Comparison of "Issuing work done" for higher and lower error rates (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

While the red graph shows the initial situation, the green graph depicts the total output of issued parts to line with 20 % higher error rates and the blue one with 10% lower error rates. According to the process experts the movement of the results for the modifications closely approximate the expected real effects. Following the successful verification tests, the actual simulation of process alternatives could be started.

Simulation of process alternatives

Table 3-4 lists the seven simulated allocations of persons to the process steps. The selection of the allocations was based on the boundary conditions given by the on-site conditions: receivers, pickers, and team leaders must be more than 2 people and line runners more than 3 people. Each option comprises 17 allocated workers.

Table 3-4 Simulated options of allocation of persons to process steps (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

	Pickers	Team leaders	Receivers	Line runners
Init. sit.	5	3	3	6
Opt2	4	3	5	5
Opt3	4	3	4	6
Opt4	5	3	4	5
Opt5	4	4	4	5
Opt6	4	3	6	4
Opt7	3	3	7	4

The simulation results illustrated in Figure 3-19 show that Opt2 offers the highest amount of issued parts. The simulation results also show that the receiving rework cycle is the most critical process step. Reducing the number of pickers by two is likely to favor the receiving team. Simultaneously, the receivers become accountable for binning the parts. This means that the whole line team is able to issue the same number of parts to the production line at a faster pace.

In the initial situation, there was always a delay between receiving the parts and binning the parts, because the receiver had to inform the pickers that a particular part was ready for binning. However, sometimes the pickers were busy with other duties. This delay was avoided in Opt2, since there is no delay between receiving and binning.

Opt4 and Opt5 demonstrate that supporting the receiving team with just one additional worker and simultaneously giving them the duty of binning would not add significant benefit. The simulation implied that to improve the initial situation, it would be necessary to support either the receiving team or the line runners. Opt4 and Opt5 differ in the number of the pickers, team leaders, and receivers. The number of receivers was raised to 4. This didn't seem to be enough to significantly change the overall result of the simulation: The need for additional workers at the receivers' workstation or the line runners' workstation. Opt6 and Opt7 were stressing the importance of the line runners and pickers. As mentioned before, there are boundaries for the optimization. If there would be less than 4 pickers, the early stage of the process would have underperformed. At the same time, it was not possible to reduce the number of line runners to less than 5, since this was also a critical process step.

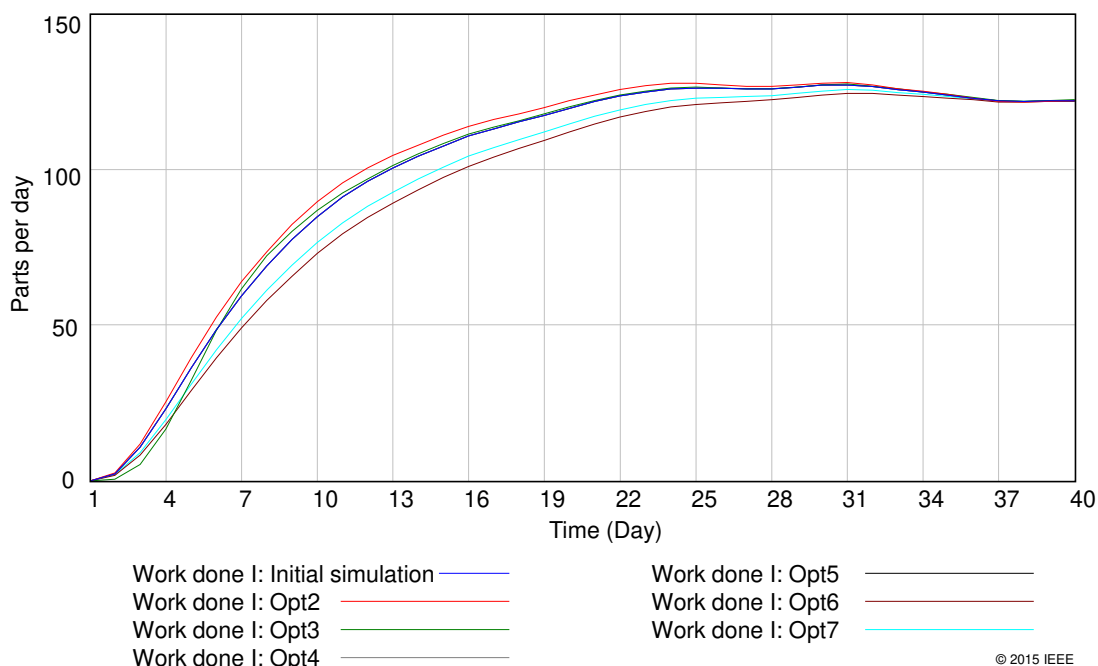


Figure 3-19 Simulation results of issued parts for the simulation allocation scenarios (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

Within the analysis part of this case study an allocation of persons to the process steps could be found that, according to the simulation, would increase the effectiveness of the process.

3.3.6 Stage 5: Disseminate

Based on the analysis results, the industry partner decided to implement the suggested process change. This offered the possibility to assess the process before and after the change and thereby measure the effect of the change implementation. For measuring the impact of the adjustment the Warehouse Performance Index (WPI) was defined. As given in Table 3-5, the WPI was calculated as sum of several ordinal values.

Table 3-5 WPI measurement (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

Affected area	Topic	Measurement
Receiving and Binning	Housekeeping receiving area <ul style="list-style-type: none"> Parts / no parts on the floor FiFo sticker on every part 	1 – 10 points
	GRVed vs. received	For every part received but not GRVed -1 point
Picking	Housekeeping picking in advanced area	1 – 10 points
	Picking in advance three days ahead and completely picked	+1 per complete unit
Issuing	Issued system wise (only when bus is completely issued before the unit goes offline)	+2 per unit
	Shortages caused by warehouse	-1 per part

The surveying of the WPI started two weeks before the change implementation.

Figure 3-20 shows the daily graph of the WPI: The days leading up to December 13th represent the WPI before the process change (green period). The average of the WPI in this period is 8.6 with a root mean square deviation of 2.4. The blue period starting December 14th represents the break at the turn of the years, where the plant was shut down. Due to the relatively high level of confusion among the workers, the first week after the change is not analyzed in greater detail to avoid measuring settlement effects. This period is handled as a learning phase and thus, it is not considered. Also the production started a week before the logistic department and thus, the line process, with the result of a fully packed receiving area at the beginning of the evaluation phase. Deliveries from the suppliers were received before that. So starting from January 13th, the WPI represents the process performance after the process change (orange period). Up to January 31st, the average WPI is 9.6 with a root mean square deviation of 4.0. The increase in WPI indicates an improvement in the line process at the industry partner. The simulation's predicted increase in the outcome of work packages (up 3.4%) is reflected in an upsurge of the WPI. It was not possible to measure the daily delivered parts. Therefore, the WPI is considered to compare the trend of the simulation with the performance of the persons in reality.

Feedback from face-to-face interviews with the workers was similarly positive. The receiving site and the pickers in particular benefited from the process change. The receivers especially pointed out their improved overview of the parts on their site. The shortage handling improved too, while the receivers now have sufficient workforce for their process step. If there is communication between the production line and the receivers, they can identify an incoming critical shortage immediately and deliver it directly to the line. This increased the output of the overall process. The delivery to the line was done in the past by line runners, due to a lack of workforce of the receivers. The pickers stated that they are pleased to have enough time to pick accurately. By reducing the headcount in this team, the team's internal communication and distribution of tasks was improved.

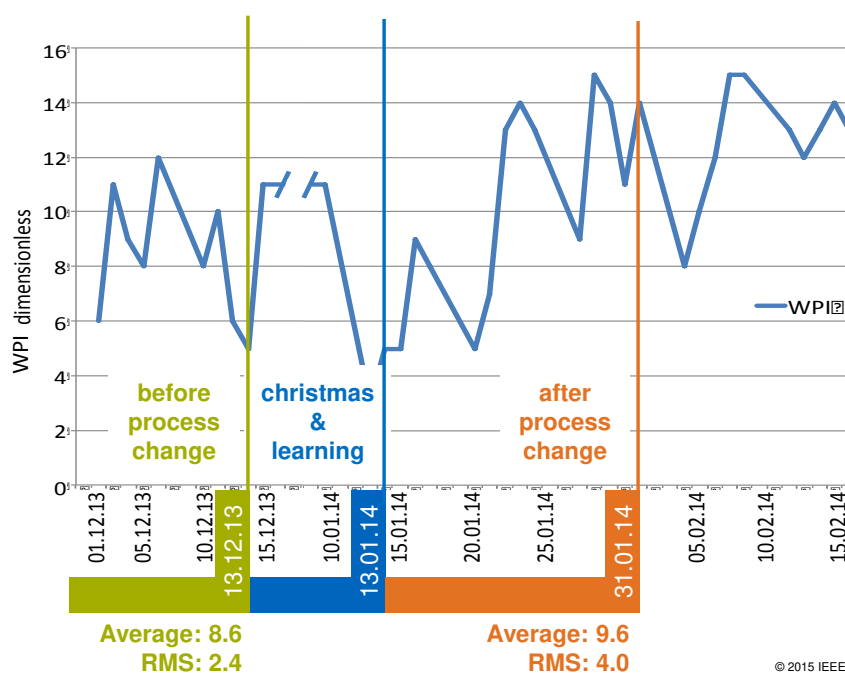


Figure 3-20 Measured WPI over time showing the effect of the implemented change (Kasperek, Bermond, Maisenbacher, Zaggl, et al., 2015)

Referring back to the initial research question of this case study, the concept of structure-based System Dynamics Analysis can be applied to line processes. However, especially the System Dynamics model has to be adapted to the differing characteristics of line processes such as the physical flow of parts through the process.

3.3.7 Implications of the case study

Detailing of the framework concept

Even though this case study investigated a line process and not an engineering design process, the general steps of the framework concept derived from case study 1 can be further detailed. Figure 3-21 illustrates the general steps of the framework concept from case study 1 on the left side and the more detailed version of the framework based on the additional insights from case study 2 on the right side.

The step transformation describes the transformation of the MDM into a System Dynamics model. The System Dynamics model can be seen as system of equations. The transformation as such can be divided in two sub steps:

The transformation of the MDM into a qualitative System Dynamics model and the quantification of the qualitative System Dynamics model. Within the first sub step the information from the MDM is used to build up the general setup of the System Dynamics model with its stock and flow constructs, variables and relations. Seeing the System Dynamics model as a system of equations, here, variables for each particular equation are defined. The subsequent sub step of quantification populates the System Dynamics model with additional

information which is necessary to enable a simulation of the engineering design process. Seeing the System Dynamics model as a system of equations, here, the form and parameters of the equations are defined. If not all domains of the MDM are transformed into the qualitative System Dynamics model, the information of the not transformed domains and their elements may be used as input for the quantification of the System Dynamics model.

After the quantification of the System Dynamics model it can be simulated. However, the model should be tested before the simulation-based analysis of the engineering design process. This includes reviewing if the model sufficiently enough represents the real engineering design process for the purpose of the analysis. Consequently the behavior of the engineering design process based on its structure represented within the MDM can be analyzed by System Dynamics simulation. After the analysis the results need to be transferred back on the structure of the engineering design process.

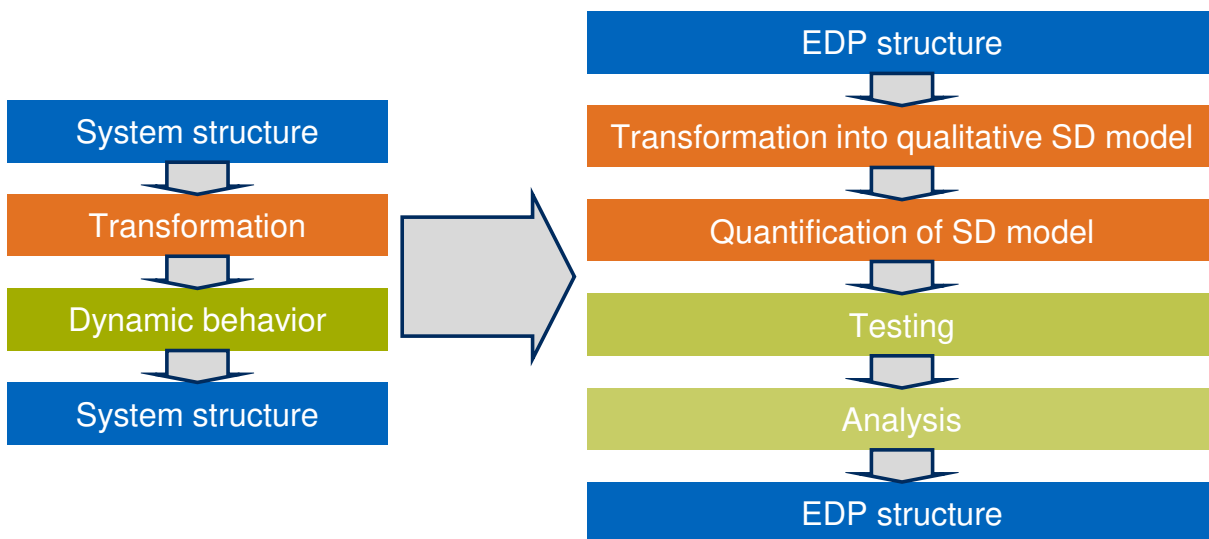


Figure 3-21 Detailed framework concept

Refining the model transformation concept

Additionally to the detailing of the framework, the model transformation concept was refined. In contrast to the first case study the transformation into rework cycles was conducted on element level. This was necessary because of the different structure of the developed MDM. While within the MDM of the first case study the domains represented the different phases of the engineering design process, the MDM of this case study depicted a more differentiated view on the structure of the line process. Thus, the transformation of the MDM was conducted on element level. This offers the possibility to not only include process steps but also to add various other components of the process into the consideration. Thereby, the transformation on element level offers a higher degree of freedom for the configuration of the model. However, a guideline is needed which domains respectively its elements can be transformed into which System Dynamics constructs as well as which domain may be input for quantification or do not necessarily be modeled for the purpose of structure-based System Dynamics Analysis. Based on the previous case studies especially the following components offer potential for the desired analysis:

- process steps
- involved persons
- required resources
- quantification information for process steps.

In accordance with the model transformation concept, process steps should usually be transformed to rework cycles (if they contain manual work), while information about involved persons and required resources can usually be transformed to forms of resource pools. Additional information especially about process steps may be used as input for the model quantification.

Guidance for model testing

Before the developed System Dynamics model can be used for analysis, it needs to be tested that the simulation model but also the underlying MDM represent reality sufficiently enough. Within the framework for structure-based System Dynamics Analysis (chapter 4), guidance should be available on how to verify the generated System Dynamics models.

3.4 Case study 3: Benchmarking process analysis at MAN Truck & Bus AG

The case study was conducted in form of a student project (Produktentwicklung, 2014c) and the herein presented results base on the publication of Kasperek, Berger, et al. (2015).

3.4.1 Description of the case

To ensure its own compatibility MAN Truck & Bus AG constantly triggers product and process improvements by benchmarking itself with competitors. For this particular case study the industry partner was interested in a process comparison of its own product benchmarking processes with product benchmarking processes of other automotive OEMs. Tools and methods of the other OEMs should be identified and analyzed. Based on a comparison with the own tools and methods currently in use, potentials for further process improvements by integrating specific tools and methods of the other OEMs should be identified.

Based on existing descriptions of the MAN benchmarking processes MDMs provided the basis to develop System Dynamics models of the MAN product benchmarking processes. The System Dynamics models, in turn, provided a simulation environment for the purpose of analysis and optimization by enabling the possibility to incorporate the others' tools and methods within the simulation.

3.4.2 Stage 1: Research question

From the perspective of the industry partner the challenge of the case study can be stated as: “How can potentials for process improvements of the own benchmarking processes be identified by integrating specific benchmarking tools and methods of the other companies?”

This allowed addressing the general question of interest from the research perspective:

“Can structural models of benchmarking processes in form of Multiple-Domain Matrices, be used as a basis to simulate the dynamic behavior of these processes by System Dynamics?”

If this question can be affirmed, the behavior of the benchmarking processes towards integrating specific tools and methods of other companies can be assessed. This would allow answering the challenge as stated by MAN Bus & Truck AG and may to identify additional implications for the structure-based System Dynamics Analysis framework.

3.4.3 Stage 2: Instrument development

The case study had a time span of six months. The instrument development differs between MAN’s own product benchmarking processes and the product benchmarking processes of the other automotive OEMs. For the MAN product benchmarking processes, protocolled interviews with the affected managers about process steps, tools and methods were the primary data basis. For the product benchmarking processes of the other automotive OEMs less or no process documentation was initially available for MAN. Based on initial contacts provided by MAN at least one protocolled interview was conducted with benchmarking experts from each OEM by the researchers. To be aware of the requested information and to prepare potential necessary documents, the participants received the questions of the protocolled interviews at least one week before the interview. Additionally, the participants were asked to show existing documentation at the interview appointment as far as possible under the given confidentiality agreements.

The questions for the protocolled interviews of the automotive OEM’s were grouped in three areas:

- general questions about the OEM’s product benchmarking division: Allocation within the company, subdivisions and employees, work portfolio
- information about the existing processes: Detailed view on the particular benchmarking processes
- general information about the work context: Further use of the results, specific examinations for special purposes, knowledge management.

To give the participants a feeling for the desired form and degree of detail of their answers, an exemplary answer of the MAN division was presented for each question.

3.4.4 Stage 3: Data gathering

Product benchmarking processes and existing tools and methods within MAN

At MAN Truck & Bus AG three major product benchmarking processes exist which are referred to as A, B, and C within the case study. A is focusing on free available information about competitors. B is focusing on automotive characteristics of the products of the competitors. C is focusing on detailed characteristics of particular parts of the automotive. Figure 3-22 shows the flowcharts of the three product benchmarking processes.

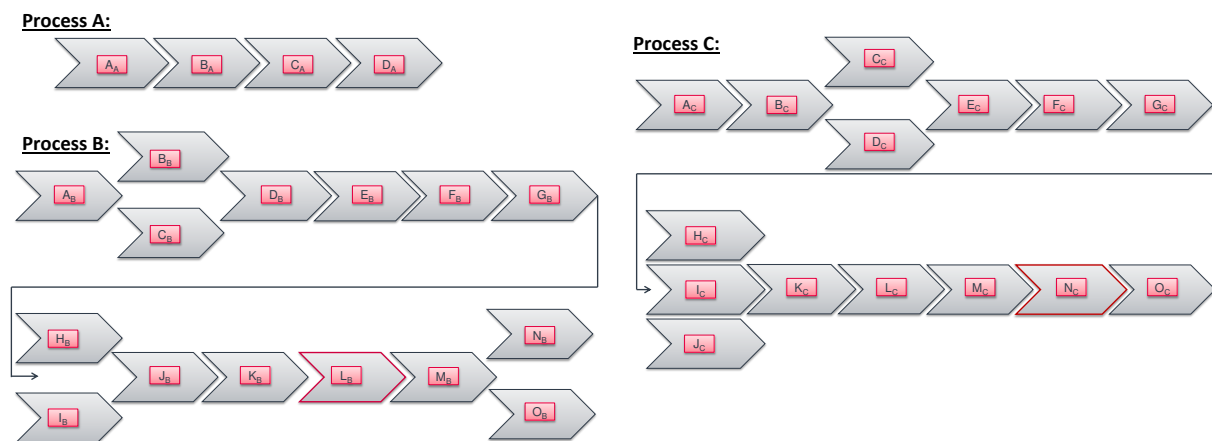


Figure 3-22 Flowcharts of the three product benchmarking process of MAN Truck & Bus AG (Kasperek, Berger, Maisenbacher, Lindemann, et al., 2015)

The three processes are of different size and have different workloads. Especially process B is also strongly influenced by other processes and divisions, so that various accompanied processes have to be taken into account when modeling the process sequence. Each step of the three processes was documented in detail by a written description of activities. Additionally organizational units involved in the particular process steps of the three processes as well as used tools and methods were documented within the interviews. Figure 3-23 shows the allocation of the organizational units, tools and methods to the particular steps. Furthermore tools and methods are used to achieve the benchmarking objectives in an efficient and effective manner. Exemplary tools and methods are fact sheets to communicate results or test drive opportunities offered to important stakeholders. While all other elements of the Organizational Units domain refer to one particular person, element E refers to a varying amount of persons. Thus, the numbers in the column of E point to the necessary number assigned to the particular process step.

		Organizational Units							Tools (shortened)										Methods								
		DM	P-I-M	P-I-G	P-I-Z	E	S	SD	Task	-	D	SP	ECL	MI	CE	SC	:	Task	PSM	DS	AC	SC	DBE	TM	MARF		
Benchmarking processes	Steps of process A	A _A	X	X					A _A	X								A _A									
		B _A		X					B _A	X								B _A									
		C _A				5 to 6			C _A				X	X				C _A									
		D _A		X		3			D _A		X							D _A		X							
	Steps of process B	A _B	X		X			X	A _B			X						A _B									
		B _B			X			X	B _B									B _B									
		C _B			X				C _B	X								C _B									
		D _B			X		X	X	D _B				X					D _B									
		E _B			X				E _B									E _B									
		F _B			X				F _B									F _B									
		G _B					2		G _B					X	X			G _B			X	X					
		H _B					3		H _B					X	X			H _B					X				
		I _B			X				I _B									I _B									
		J _B			X				J _B		X							J _B		X					X		
		K _B					1		K _B									K _B								X	
		L _B					2		X	L _B		X			X	X	X		L _B							X	
		M _B			X					M _B		X							M _B								
		N _B			X				X	N _B									N _B								
		O _B			X	X				O _B									O _B								
		Steps of process C	A _C	X			X			A _C			X						A _C								
	B _C					X		X	B _C									B _C									
	C _C					X			C _C									C _C	X								
	D _C				X			X	D _C	X			X					D _C									
	E _C					1	X		E _C									E _C									
	F _C					2			F _C				X	X				F _C			X	X					
	G _C					3			G _C				X	X				G _C					X				
	H _C				X	3 to 4			H _C				X					H _C								X	
	I _C							X	I _C									I _C									
	J _C					1 to 2			J _C				X	X	X			J _C							X		
	K _C				X	4			K _C									K _C									
	L _C				X	4			L _C				X	X				L _C									
	M _C				X	4			M _C				X	X				M _C									
	N _C					4		X	N _C		X		X	X				N _C									
	O _C					1			O _C									O _C									

Figure 3-23 Allocation of organizational units, methods and tools to process steps (Kasperek, Berger, Maisenbacher, Lindemann, et al., 2015)

Tools of the other OEMs and decision for focused tools

Based on the protocolled interviews with the process experts of the other OEMs, various tools and methods were identified for their particular benchmarking processes. Even though tools were often labelled differently within the OEMs, most tools and methods were used in related constellations. For a better overview the identified tools were abstracted to categories of tools depending on the purpose of the tool. Figure 3-24 shows the exemplary allocation of tools and methods to the categories for each company. The categories were defined in discussion and agreement with the industry partner. The tools of each company were then allocated to these categories to allow for a comparison between the companies. As some categories of tools were not in the area of responsibility of the industry partner these categories were indicated and not further considered.

Identified Tools	MAN Truck & Bus	OEM 1	OEM 2	OEM 3	OEM 4
Category 1	Tool_A	Tool_D	Tool_G	Tool_D	Tool_D
Category 2	Tool_B	-	Tool_H	Tool_K	Tool_B
Category 3	Tool_C	Tool_E	-	Tool_I	-
Category 4	Implicitly existing	Tool_F	-	Tool_J	-

Figure 3-24 Exemplary allocation of tools to categories for each OEM (Kasperek, Berger, Maisenbacher, Lindemann, et al., 2015)

Based on the comparison, particular tools and methods of interest were selected by the industry partner based on the three categories:

- conformance with process optimization targets
- possible time to implementation
- transferability to MAN context.

MDM modeling

Based on the implications of the previous case studies and the metamodel for engineering design processes of Kreimeyer (2009, p. 111), the domains of the MDM were chosen as indicated in Figure 3-25. The initially suggested resource domain of Kreimeyer (2009, p. 111) was detailed into tools and methods due to the specific conditions of the case study.

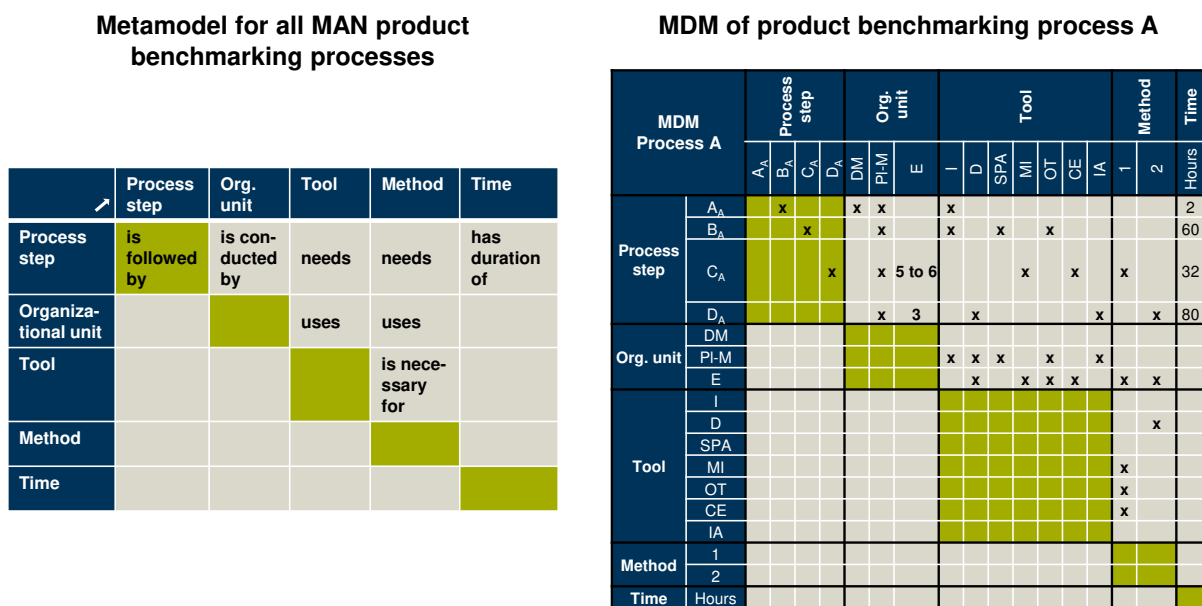


Figure 3-25 Metamodel of the product benchmarking processes (left) and MDM of process A (right) (Kasperek, Berger, Maisenbacher, Lindemann, et al., 2015)

As suggested by Kreimeyer (2009, p. 111), process steps (called tasks at Kreimeyer (2009)), organizational units and time were chosen as domains. A detailed description of the domains can be found in Kreimeyer (2009). Based on this metamodel the detailed MDMs for the particular processes A, B and C were created. The process step sequences were modeled based on the previously developed process models, while the organizational units, tools, and methods were assigned based on the results of the protocolled interviews. The right side of Figure 3-25 shows the detailed MDM for process A. The metamodels of the processes B and C as well as the corresponding MDMs are not illustrated due to confidentiality reasons.

3.4.5 Stage 4: Analyze Data

Especially for process B and C the granularity of process steps was high. After consulting the industry partner particular sequences of process steps were concentrated to one aggregated step with respect to the desired focus of the study. For eased System Dynamics modeling not all dependencies indicated within the metamodel of the MDM were incorporated in the System Dynamics models: According to Maurer (2007), particular subsets of MDMs can be deduced indirectly by matrix multiplication. The subsets *Org. unit uses Tool*, *Org. unit uses Method* as well as *Tool is necessary for Method* were not directly modeled in System Dynamics as the corresponding information was already incorporated indirectly by modeling the following other subsets *Process step is conducted by Org. unit*, *Process steps need Tools* and *Process steps need Methods*. The underlying assumption for modeling *Org. unit uses Tool* (and correspondingly for the other two subsets) indirectly is: If a *Process step is conducted by an Org. unit* and also a particular *Tool is needed for this Process step*, then it does not have to be modeled which *Org. unit needs which Tool* as *Org. units* and *Tool* for the particular *Process step* are already defined.

If the subsets are indirectly deduced, more relations are calculated than actually really exist according to the documentation. This is due to the fact that the calculation shows all potential relations and this number differs from the measured number of real relations. Consequently the meanings of these calculated subsets are slightly different: *Org. unit might uses Tool*, *org. unit might uses Method* as well as *Tool might be necessary for Method*. The mapping of the number of calculated relations to the measured number of relations offered values from 50% to 80% with a mean of 64% and a standard deviation of 11%. The benefit of the calculated dependencies is that they do not have to be directly modeled within the System Dynamics. For this case study the effect of eased modeling was weighted higher than the loss of accuracy of the model.

Qualitative System Dynamics modeling

For this case study, the transformation method of case study 2 was used as a basis. Each process step respectively aggregated process step was transformed into the System Dynamics composite construct of a rework cycle. In particular the rework cycle construct of Reichelt & Lyneis (1999) was used. Additionally the variable *Start task* was defined to enable coordination between the rework cycles by modeling the following condition: A subsequent rework cycle can only start if a particular amount of work packages has already been processed error-free, thus reached the *work done* stock. To increase the comprehensibility of the System Dynamics

model, a composition panel for the allocation of organizational unit, tools and methods to the particular process steps was built. Within this panel the organizational unit, tools and methods are bundled to *Allocated Org. units*, *Used Tools* and *Used Methods* for each particular process step.

Figure 3-26 shows the qualitative System Dynamics model for process A. The other processes of the industry partner were modelled in a similar way.

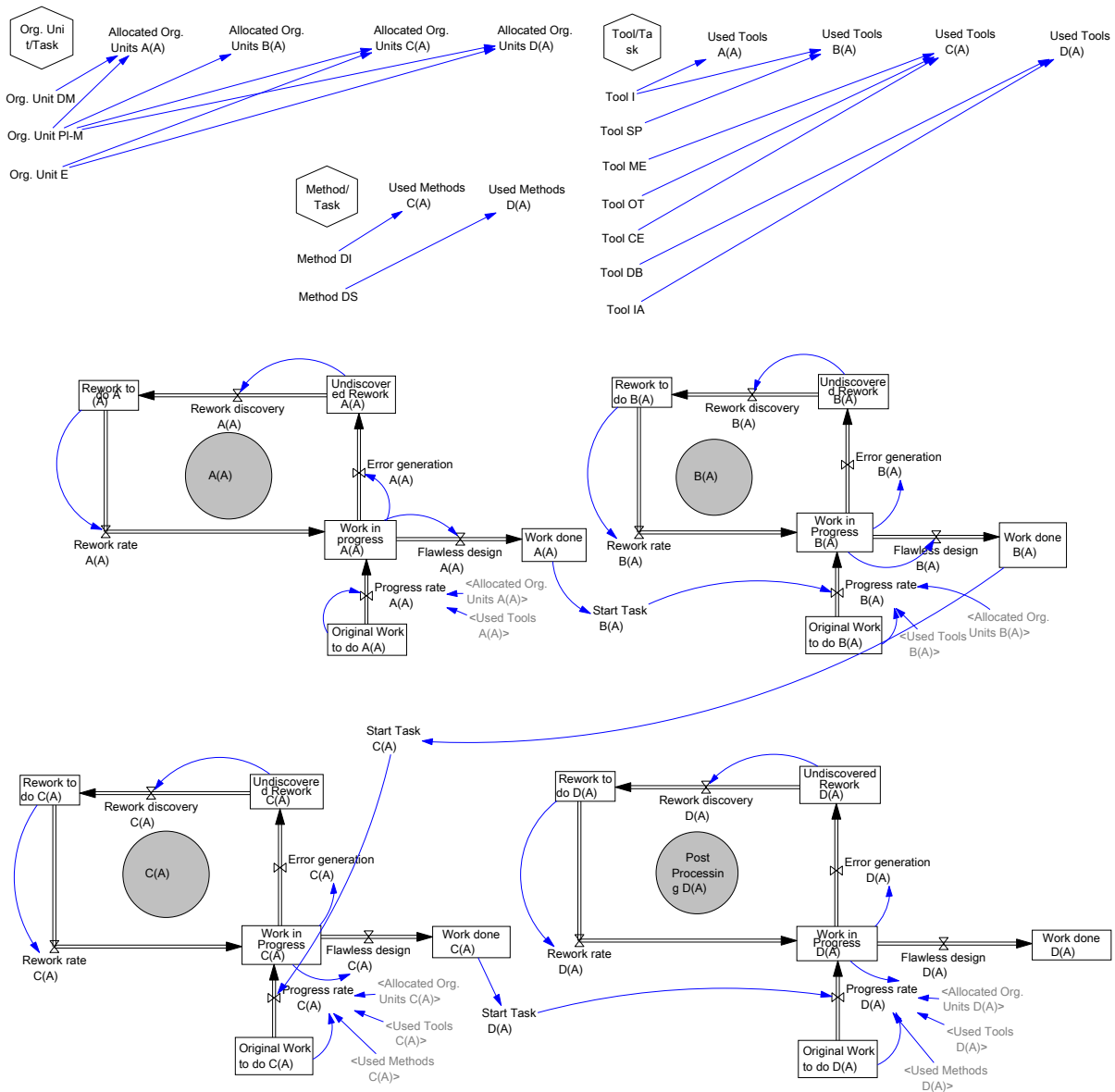


Figure 3-26 Qualitative System Dynamics model of product benchmarking process A (Kasperek, Berger, Maisenbacher, Lindemann, et al., 2015)

Quantitative System Dynamics modeling

As a first step of quantification the initial values of each stock were defined as 0, except of the *Original Work to do* variables which represent the original amount of work packages for each process step. This amount was normalized and set to 100.

As far as equations were implemented within the models for the quantification, the same types of equations were used for each of the three MAN processes (though, with customized values for each process). The different values of the duration, the progress rate and the general amount of occurring rework for each processes step of all three processes were estimated based on interviews and workshops. Within Table 3-6 the different equations implemented within the System Dynamics model are given.

Table 3-6 Equations implemented within the System Dynamics model (Kasperek, Berger, Maisenbacher, Lindemann, et al., 2015)

Variable	Equation	Unit
Original Work to do	= - Progress rate	WP
Progress rate	= IF THEN ELSE ("Start Task" = 1 :AND: "Original Work to do" – (100/T)*0.0078125 >= 0 :AND: "Allocated Org. Units" > 0 :AND: "Used Tools" > 0 :AND: "Used Methods" > 0, 100/T, 0)	WP/t
Work in Progress	= + Progress rate + Rework rate – Flawless design – Error generation	WP
Flawless design	= x * Work in Progress	WP/t
Error generation	= (1 – x) * Work in Progress	WP/t
Work done	= + Flawless design	WP
Undiscovered Rework	= + Error generation – Rework discovery	WP
Rework discovery	= 0.8 * Undiscovered Rework	WP/t
Rework to do	= + Rework discovery – Rework rate	WP
Rework rate	= Rework to do	WP/t
Start Task	= IF THEN ELSE ("Work done" > 90, 1, 0)	/

The values of the progress rate were calculated by dividing the normalized *Original Work to do* by the estimated duration of the particular process step.

Simulation of the engineering design process

Figure 3-27 shows exemplary simulation results for the first process step of product benchmarking process A. As the same equations; even though with different implemented values, were used for each process step the principal characteristics of the graphs of this process step give an overview of the general simulation results. Within Figure 3-27,

a) shows the trends of *Original Work to do*, *Progress rate* and *Work in Progress*. The *Original Work to do* decreases with a certain *Progress Rate* and finally reaches 0. In the meantime *Work in Progress* increases and reaches its maximum when all original work has been done. Afterwards there is still work in progress due to rework, but decreases slowly.

b) shows the trends of *Work in Progress*, *Flawless Design* and *Work done*. It can be seen that the trend of *Flawless Design* is scaled based on the trend of *Work in Progress* and can be also seen based on the corresponding equation indicated in Table 3-6. The difference indicates the amount of work packages which still have to be reworked (*Error generation*). The maximum of *Flawless Design* indicates the inflection of *Work done*.

c) shows the trends of *Work in Progress*, *Error generation* and *Undiscovered Rework*. It can be seen that the trend of *Error generation* is also scaled based on the trend of *Work in Progress*. The *Error generation* also influences the *Undiscovered Rework*. Thus, it starts to decrease shortly after the *Error generation* reaches its maximum.

d) shows the trends of *Rework discovery*, *Rework to do* and *Undiscovered Rework*. The *Rework discovery* indicates 80% of the *Undiscovered Rework* due to the assumption that 80% of rework are discovered immediately. The *Rework to do* indicates the amount discovered rework minus the amount of already reworked work packages.

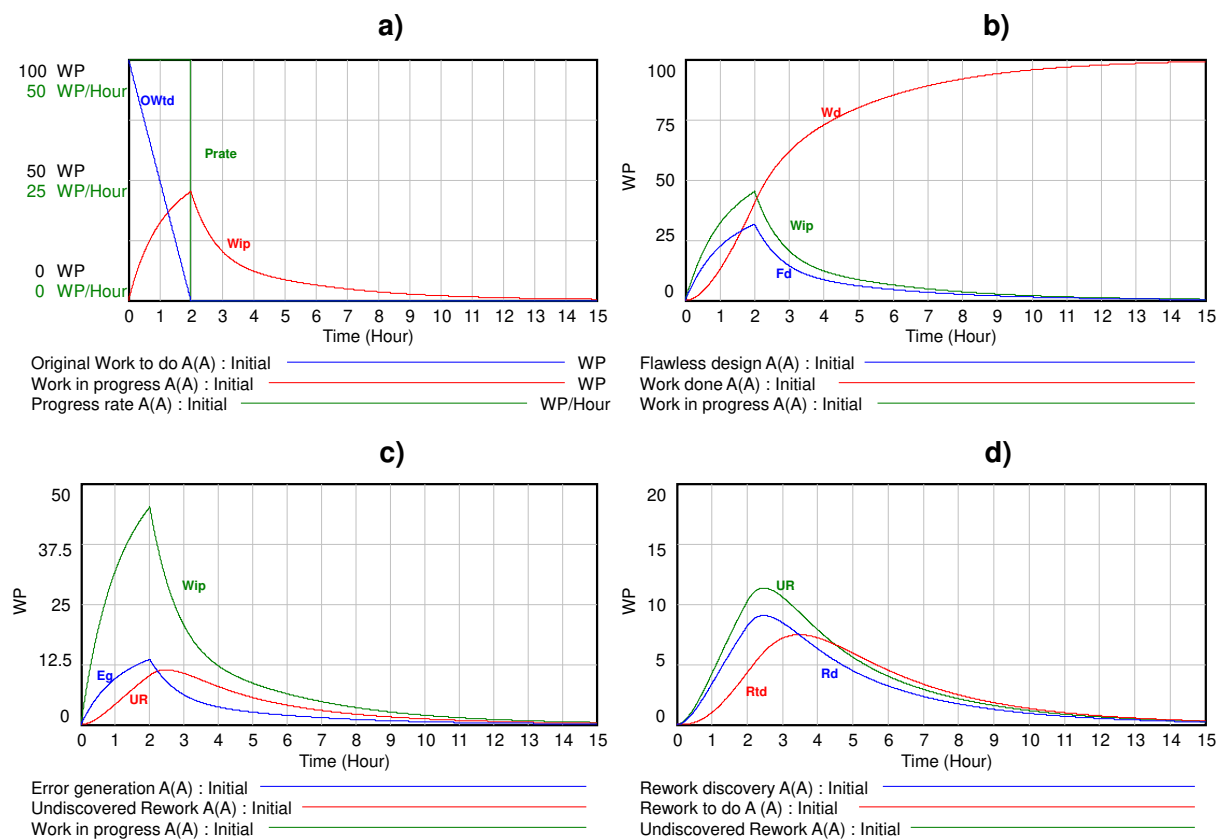


Figure 3-27 Exemplary simulation results of the first process step of benchmarking process A (Kasperek, Berger, Maisenbacher, Lindemann, et al., 2015)

To verify the simulation results of the three processes the results of each simulation were discussed and agreed on with the responsible MAN manager. Within a workshop it was shown which input from whom was incorporated in the model. After it was approved by the management that the models sufficiently enough represent the real processes, the model can be further used to identify optimization potentials by incorporating tools and methods from the benchmarking partner within the own process simulations.

Process simulation with additional tools and methods

Based on the developed simulation of the existing product benchmarking processes B, C and the identified tools and methods of the other OEMs, the effects of the use of these tools and methods within the MAN processes were simulated to identify optimization potentials. The product benchmarking process A was not considered for the simulative analysis as no additional tools and methods could be identified within the product benchmarking processes of the other OEMs.

The influence of the additional tools and methods on the processes B and C was estimated within workshops with the corresponding managers. Within these workshops it was decided which parameters would be influenced by the particular tools and methods, as well as how this influence would look like. This information was taken as a basis to simulate the influence of each additional tool and method. In most cases additional tools and methods were supposed to decrease the error rate within existing process steps.

The optimization of the benchmarking processes had two goals: increasing the quality of the process outcomes as well as decreasing the process time. These two goals were to a certain extent contrary as additional tools and methods for increasing the quality of outcomes often also increase the process duration. Based on these goals three possible outcomes for the incorporation of additional tools and methods with respect to the process duration were identified:

- tools and methods decreasing the duration of a particular process step by decreasing the error rate
- tools and methods increasing the duration of a particular process step by increasing the amount of work to do
- tools and methods increasing the overall process duration by adding an additional process step.

Table 3-7 shows the simulated process step durations of process B. Process steps where additional tools and methods were implemented are indicated in yellow. The left part of Table 3-7 shows the initial simulation values, the right part shows the simulated values after the addition of selected tools and methods. The very right column shows the delta before and after the addition of tools and methods. The values for cells indicated with “...” were simulated, but are not presented in this thesis due to confidentiality reasons.

Table 3-7 Simulated values of process step durations of process B before and after the simulated addition of tools and methods (Kasparek, Berger, Maisenbacher, Lindemann, et al., 2015)

Whole Vehicle Process							
Task	Initial simulation values			Simulation values after addition of the selected tools and methods			Delta
	Point in time of "Start Task" [Weeks]	Point in time of "Work done" >=90 [Weeks]	Time per Task [Weeks]	Point in time of "Start Task" [Weeks]	Point in time "Work done" >=90 [Weeks]	Time per Task [Weeks]	Expenditure of time [Weeks]
A _B	0	3,46094	3,46094	0	3,46094	3,46094	
B _B	3,46094	15,1016	11,64066	3,46094	15,1016	11,64066	
C _B		13,2109	9,74996		11,0234	7,56246	-2,1875
D _B	15,1016	38,8281	23,7265	15,1016	38,8281	23,7265	
E _B	38,8281	45,8984	7,0703	38,8281	45,8984	7,0703	
F _B	45,8984	48,7422	2,8438	45,8984	48,7422	2,8438	
G* _B	48,7422	55,9063	7,1641	48,7422	56,5313	7,7891	0,625
J' _B	/	/	0	56,5313	59,0938	2,5625	2,5625
K _B	55,9063	65,7344	9,8281	59,0938	64,7109	5,6171	-4,211
L _B	
M _B	4,5	-0,3984
M' _B	2,5625	2,5625
N _B	
O _B	
Time for the whole process				

3.4.6 Stage 5: Disseminate

The incorporation of additional tools and methods offers several optimization potentials for MAN's benchmarking processes with respect to duration and quality. The potentials with respect to process duration were assessed based on the simulation results. Therefore, as shown in Table 3-7, the simulated process durations with and without the additional tools and methods were compared. The industry partner based its decision which tools and methods to incorporate on their influence on process duration as well as on quality aspects. A scheme was developed to show both aspects within one view. Table 3-8 shows the corresponding tools and methods sheet.

This sheet consists of the process steps, the potential additional tools and methods, the simulated difference in time between process step duration with and without the additional tool or method and the assumed effect on the quality. To allow a quick overview the simulated duration delta was also indicated with a color scale from red (significantly increased duration) to green (strong time saving).

The effect on quality was assessed based on the results of interview series with the involved managers and rated from (--) for an expected significant quality decrease to (++) for an expected

significant quality increase. The particular lines of argumentation of the management for an expected quality increase or decrease were also documented. Based on the tools and methods sheet and the lines of argumentation, suggestions for which tools and methods to incorporate within the MAN processes on a short-term, medium-term and long-term perspective were given by the researchers.

Table 3-8 Tools and methods sheet for management decision with exemplary tools and methods (Kasperek, Berger, Maisenbacher, Lindemann, et al., 2015)

Whole Vehicle process			
Process step	Tool / new step	Time delta [Weeks]	Quality increase
C_B	Tool_ISC	-2,1875	+
M_B	Tool_C	-0,3984	o
G*_B	Tool_C	0,625	o
K_B	Tool_VPNP	-3,5156	++
J'_B	new task	2,5625	++
M'_B	new task	2,5625	++
K_B	Tool_RIA	-0,6954	+
Whole Time [Weeks]:		-1,0469	+

The tools and methods sheet was accompanied by an additional document indicating the argumentations why particular tools and methods have an effect on the overall quality of the processes.

Referring back to the initial research question of this case study, the concept of structure-based System Dynamics Analysis can be applied to benchmarking processes. During this case study the concept of structure-based System Dynamics Analysis could be significantly enhanced.

3.4.7 Implications of the Case Study

The implications of this case study are driven by the insight that especially the high degree of modeling freedom, which is also the strength of MDM and System Dynamics modeling, brings challenges in finding the best way of modeling and ensuring the reproducibility of the model.

MDM metamodel for the structure-based System Dynamics Analysis

Comparing the qualitative System Dynamics models of the first three case studies shows that their structure offers a high degree of similarity. Common domains in the context of structure-based System Dynamics Analysis seem to exist. Furthermore, the dependencies between the elements of these domains can be modelled in a similar manner. Previous work in this field exists in the form of the metamodel for engineering design processes by Kreimeyer (2009, p. 111). However, this metamodel does not completely fit to the purpose of a structure-based System Dynamics Analysis and therefore, needs to be adapted.

Guidance for when to use which rework cycle

Within the previous case studies different rework cycles were applied. Existing System Dynamics literature offers a variety of different rework cycles concepts. Furthermore, these original concepts have been customized to specific conditions various times by innumerable authors. As literature lacks an overview of all rework cycle concepts and their adaptations, often the original concepts are chosen and customized again without incorporating existing expertise from literature. Therefore, a guideline in form of a rework cycle adaption scheme is necessary.

Reproducibility of quantification process (Quantification sheets)

The case study revealed that the capturing and documentation of information has to be supported. The MDM and the System Dynamics model are currently the only locations where the generated data can be stored. While the data stored in the MDM is usually easily accessible, especially the data and assumptions used for the quantification are well hidden within the equations behind the System Dynamics model. However, within the previous case studies this information was identified as very important and should be reproducibly documented. Even though most of modellers have an aversion against documentation (see Kasperek et al. (2015) for a review), documentation is important: Without documentation it is difficult to decide for others than the modelers themselves if the structural model at hand is sufficient in quality, scope and underlying information for the desired purpose (Kasperek, Maisenbacher, Kohn, et al., 2015).

3.5 Case study 4: CALA construction process

3.5.1 Description of the case

The Center for Advanced Laser Applications (CALA) - a joint research project between the Ludwig Maximilian University of Munich and the Technical University of Munich - is a new 70-million-Euro laser center expanding on the already existing, broad range of cutting-edge research into laser science and technology for applications in the fields of life sciences and medicine in the Munich research area. (Naeser, 2015)

The case study is concerned with the supply and integration of the laser modules at the construction site and the subsequent commissioning of the CALA. Due to the composition of the CALA and its system elements which are specifically designed for the novel techniques pursued in CALA and the fact that there is only few knowledge available on how to construct facilities for such basic research, project management desired a simulation environment for risk analysis.

The available time span for the supply and integration of the laser modules and the commissioning was limited to 1.5 years. The specific challenge expected by project management was the logistics of the process. Due to the present uncertainty at the construction site, just-in-time delivery was not an option. On the other hand storage space was limited, too. To avoid storage overflow and also mutual blocking of the bulky modules within the storage, the delivery of the modules needed to be coordinated. Additionally the availability and capacity

of the human resources had to be taken into account as well as numerous dependencies of the process steps.

The supply and integration of the laser modules at the construction site and the subsequent commissioning are a complex process where various previously undetected defects can take effect. Even relatively small defects could lead to over proportionally large delays if severe consequential damages occur or deadlines are exceeded. Overall the process and its environment are subject to various uncertainties and imponderables. Therefore, the goal of the case study was to develop a risk robust concept for the sequence of this process by assessing the impact of different scenarios on the process.

3.5.2 Stage 1: Research question

From the perspective of the responsible project managers the question of interest could be stated as:

“How should the process from supply and integration of the laser modules to the commissioning of the CALA facility be optimally structured with respect to logistics and resource allocation risks?”

This allows addressing the general questions of interest from the research perspective:

“Can the concept of structure-based System Dynamics Analysis be applied in the context of risk analysis? If so, can additional elements for the framework of structure-based System Dynamics Analysis be identified?”

3.5.3 Stage 2: Instrument development

The project team of the CALA comprised the professors who would later use the facility, respectively their representatives, the project management and external consulting firms. As this group met on a regular weekly basis, these meetings were chosen as major source for information acquisition and were attended by a researcher over six months. Due to the setup of the case study, access to most of the project management data was given. Also a work breakdown structure with sub projects and work packages was available. Several work packages of the work break down structure which comprised a similar structure especially the construction of the caves were concentrated to one domain within the MDM metamodel. As the case study was conducted in close cooperation with the CALA project management the particular intermediate results of the approach to be conducted such as the MDM and the System Dynamics model were regularly presented and discussed with the project management to verify their correctness with respect to the overall purpose.

3.5.4 Stage 3: Data gathering

Based on the experience of the previous case studies, the data gathering stage of this study was subdivided into a system definition phase where the domains and dependencies of the MDM,

thus the MDM metamodel, are defined and an information acquisition phase where the MDM metamodel is populated with detailed data and additional data for the simulation is acquired.

System definition

For the definition of the domains and dependencies during system definition the work break down structure of the process and the general spatial layout of the CALA construction site was used as a basis. Figure 3-28 shows a simplified plan view of the spatial layout of the CALA construction site. The CALA system can be divided into four major sub systems: Atlas, LBD, the Caves and the Stockyard. The construction phases for each of these subsystems were chosen as domains. Within the Atlas subsystem (indicated in red) the laser beam as primary source for the research experiments is generated and amplified. The LBD (indicated in blue) is a vacuum system through which the laser beam is guided into the experimental caves. The LBD is steeped into the ground floor of the CALA which implies that the floor has to be opened for work on the LBD and no other goods can be transported over the opened areas during the construction period. The Caves (indicated in green) represent the experimental caves of the CALA where experiments with the laser beam can be conducted. The fourth sub system of the CALA is the Stockyard (indicated in yellow). After the commissioning the Stockyard will be used for the installation of experimental setups. During the construction process the Stockyard will be used as storage for the arriving parts.

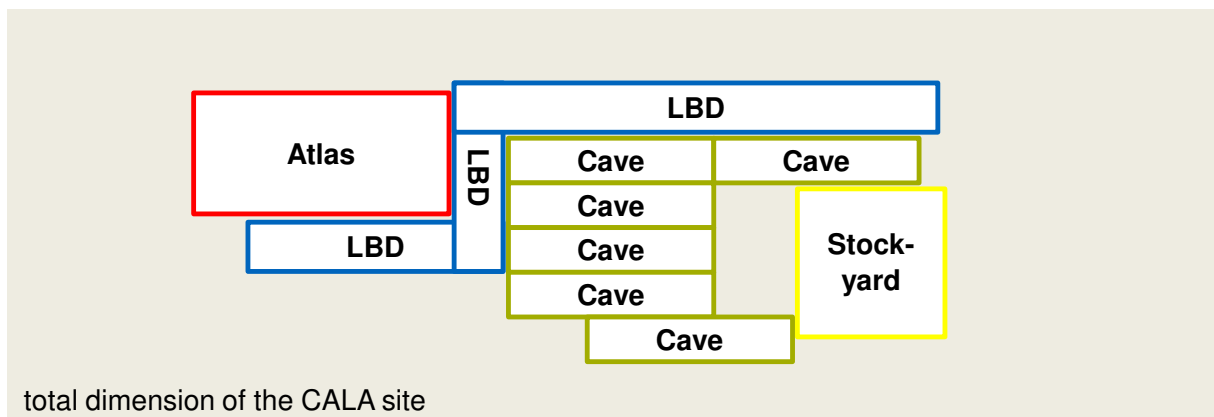


Figure 3-28 Simplified plan view of the CALA site

Figure 3-29 shows the developed metamodel of the MDM for the case study with three domains for the construction of the subsystems Atlas, LBD and Stockyard and two domains for the necessary organizational units and resources. At this stage the metamodel only referred to the mandatory dependencies of the design process and not the actual sequence to be conducted.

META MODEL CALA	Construction Atlas	Construction LBD	Construction Caves	Org. Unit	Resource
Construction Atlas				is conducted by	requires
Construction LBD	precedes temporally		precedes temporally	is conducted by	requires
Construction Caves				is conducted by	requires
Org. Unit					
Resource					

Figure 3-29 Metamodel of the MDM for the case study

Qualitative information acquisition

The structure defined by the metamodel was further detailed to derive the MDM model with all mandatory relations which could be used as input for a System Dynamics model of the process. The allocation of organizational units and resources to the particular process steps for the subsystems could be extracted from the available work break down structure. The relations with regard to the mandatory sequential relations between the constructions of the subsystem were derived based on the weekly meeting of the project team and by interviews with project managers. All other subsets were empty. Based on the information acquisition the boundary conditions indicated in Table 3-9 were identified.

Table 3-9 Boundary (BC) conditions of the CALA process

ID	Condition
BC 1	The construction of the LBD subsystem can be divided into a western and an eastern part.
BC 2	There are six caves to be constructed. One of them is located in another building.
BC 3	The subsystems Atlas and LBD need different resources. The caves 1 to 5 can be constructed using the same resources. For cave 6 specific resources are necessary. Additionally c-parts are necessary for the construction of all subsystems.
BC 4	Regarding the interfaces between LBD and Atlas respectively the Caves 1 to 5, the LBD-West has to be built before the Atlas and the LBD-East before the Caves can be set up.
BC 5	LBD-West and East have to be built up by the same team meaning that they cannot be built up in parallel.
BC 6	While team 1 is setting up the Atlas it cannot work on any other subsystem.
BC 7	The construction of caves 1 to 5 can be regarded as identical. Cave 6 already exists and does not have an interface to the other subsystems. It can be regarded as independent.
BC 8	With respect to further boundary conditions all caves except 3 and 4 have to be set up by different teams. Caves 3 and 4 will be set up by the same team.

Figure 3-30 illustrates a zoom into relevant subsets of the Multiple-Domain Matrix for CALA comprising all mandatory relations.

Multiple-Domain Matrix CALA		Construction Atlas	Construction LBD		Construction Caves						Organizational Unit					Resource				
		Atlas	LBD-West	LBD-East	Cave 1	Cave 2	Cave 3	Cave 4	Cave 5	Cave 6	Team 1	Team 2	Team 3	Team 4	Team 5	R-Atlas	R-LBD	R-Caves 1-5	R-Cave 6	C-Parts
Construction Atlas	Atlas										x					x				x
Construction LBD	LBD-West	x										x					x			x
	LBD-East				x	x	x	x	x		x						x			x
Construction Caves	Cave 1									x								x		x
	Cave 2										x							x		x
	Cave 3										x							x		x
	Cave 4											x						x		x
	Cave 5												x					x		x
	Cave 6														x				x	x

Figure 3-30 Zoom into relevant subsets of the Multiple-Domain Matrix for CALA comprising all mandatory relations

Quantitative information acquisition

As second step of information acquisition for the System Dynamics model, additionally to the structural input in form of the MDM, the necessary information for the quantification of the System Dynamics model was derived.

This includes the fixed start date for the supply and integration of the laser modules at the construction site and the subsequent commissioning of the CALA on the 06/01/2016 and the warranted date of full functionality on the 12/31/2017. There will be a two week break at the turn of the year 16/17. Taking all conditions into account the maximum overall duration of the process is 75 weeks.

The durations for the particular construction steps of the sub systems were estimated based on the existing work break down structure and expert interviews with the responsible researchers and additionally supported by the project management where necessary.

- LBD-West: 2 weeks
- LBD-East: 8 weeks
- Atlas: 12 weeks
- Cave 1-5: 50 weeks
- Cave 6: 25 weeks.

During the construction of CALA a stockyard for the incoming yet not necessitated parts is required. Therefore the hall which will be later used for the installation of experimental setups will be used as stockyard. The stockyard has a size of 180 m².

Each arriving part is assigned a date of arrival, the necessary space in m² and the rate of consumption. The arrival and consumption can either be continuous or discrete.

The parts for the LBD will arrive in five deliveries. For each delivery 10 m² stockyard space will be required. The dates of arrival are known.

The Atlas subsystems is made up of an existing laser and an extension. The existing laser does not need stockyard capacity as it is already installed in an adjoining building, where it will be dismantled and directly remounted in the CALA. The extensions will arrive in three deliveries and will be continuously consumed over the duration of the construction step. For each delivery 10 m² stockyard space will be required.

The parts for the caves 1-5 will be continuously delivered and consumed. Cave 6 already exists in an adjoining building. A specialist firm will be required for its dismantling and remounting in CALA. This will require 25 weeks and 50 m² of stockyard space.

The organizational units will be modeled as binary artifacts, to allow for an overview on which and how many operations they are involved.

Table 3-10 gives an overview of the gather data of the quantitative information acquisition.

Table 3-10 Quantified information for the construction of the subsystems

Construction of subsystem	Duration (weeks)	Delivery	Req. space in stockyard (m²)	Consumption
Atlas	12	3x	25	15 m ² continuously + 10 m ² discrete
LBD-West	2	5x	10	discrete
LBD-East	8	5x	2	discrete
Cave 1	50	continuously	2	continuously
Cave 2	50	continuously	2	continuously
Cave 3	50	continuously	2	continuously
Cave 4	50	continuously	2	continuously
Cave 5	50	continuously	2	continuously
Cave 6	25	1x	50	discrete

3.5.5 Stage 4: Analyze Data

Qualitative System Dynamics model

Based on the data gathering stage, it was decided to split the System Dynamics model in four model constructs: For the construction phases of the subsystems; for the stockyard; for the organizational units and a panel where all relevant process parameters can be seen. Figure 3-31 shows the skeleton of the System Dynamics model with these four parts. For comprehensibility reasons the arrangement of the model constructs for the construction phases of the subsystems and the stockyard was loosely based on the real physical locations at CALA. To reduce the complicatedness of the model the constructs for the organizational units were placed in a separate part of the model and linked to the other model constructs by using pointers.

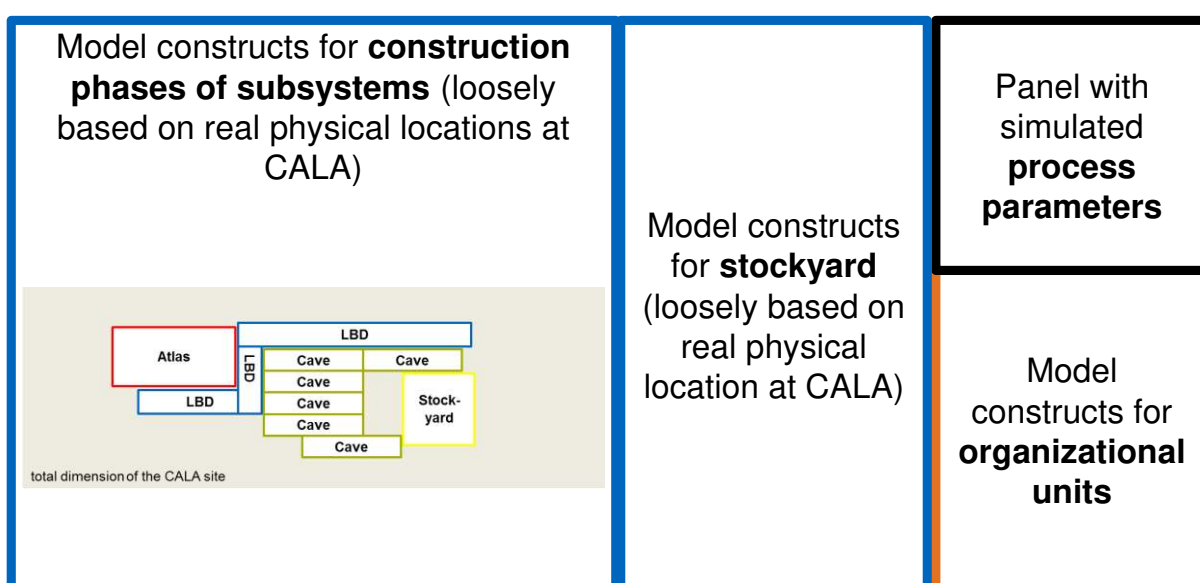


Figure 3-31 Skeleton of the System Dynamics model of the CALA construction process

The detailed System Dynamics model is shown in Figure 3-32, Figure 3-34 and Figure 3-35, each of them illustrating another part of the overall model.

Figure 3-32 shows the modeling constructs for the construction phases of the subsystems. Each construction phase of a subsystem is represented by the same modeling construct to decrease the complicatedness of the model (C 1 to C 6 represent the Caves 1 to 6). The particular differences of the construction phases were implemented during the quantification of the model. Within Figure 3-32 the colors of the arrows represent different types of dependencies. Blue represents the dependencies between variables within the System Dynamics model. Green represents Finish-Start dependencies according to the MDM in Figure 3-30. Red represents the boundary condition BC 4 and black BC 6 of Table 3-9.

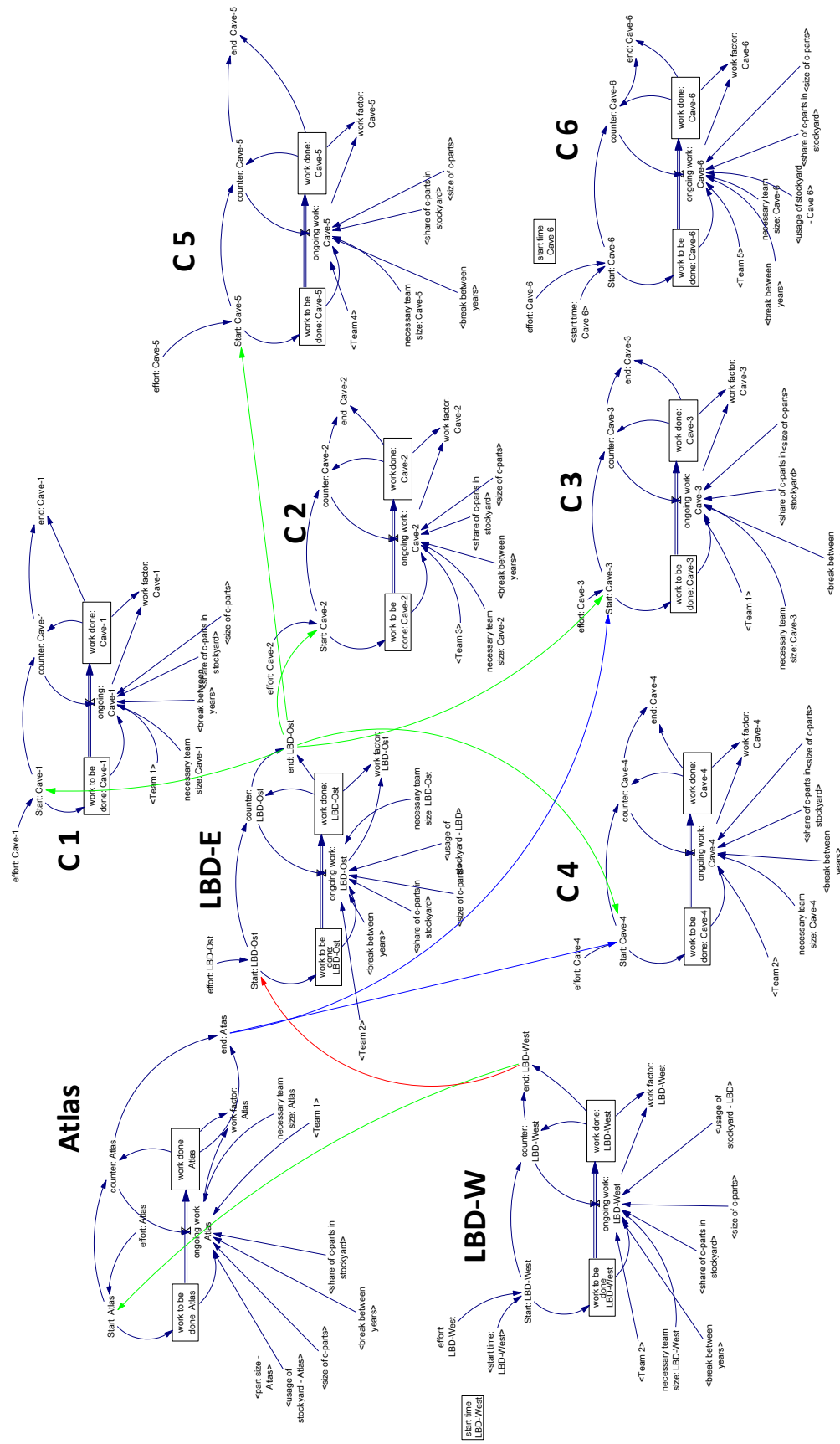


Figure 3-32

System Dynamics model: Model constructs for construction phases of the subsystems

Figure 3-33 shows the modeling construct for the construction phase of the Cave 1 subsystem on the upper middle of Figure 3-32. While for the case studies 1, 2 and 3 rework cycle concepts were used as System Dynamics modeling constructs, simpler stock-flow concepts were used in this case study due to the abstraction level and purpose of the model. Figure 3-33 illustrates the stock-flow concept with the two stocks *work to be done*, *work done* as well as the flow *ongoing work* between them.

The stocks and the flow interact with various variables: *start Cave-1* initializes *work to be done*. On the other hand it is triggered by *end: LBD-East* and its initial value is set by *effort: Cave-1*. Additionally its information is used by the *counter: Cave-1* which signals the end of the construction phase for Cave 1 by comparing *start: Cave-1* with *work done: C1*. *counter: Cave-1* stops the *ongoing work* flow and *end: Cave-1* is the inverted of *counter: Cave-1* verified with the *work done* stock. The *work factor: Cave 1* represents the amount of already spent effort for Cave 1 as long as the construction phase of Cave 1 is active. *ongoing work* transfers work from *work to be done* to *work done* if *Team 1* is at least as large as the *necessary team size: Cave-1*, it is not the break between the years 2016/17, there are still c-parts in the stockyard *share of c-parts in stockyard* for consumption *size of c-parts*.

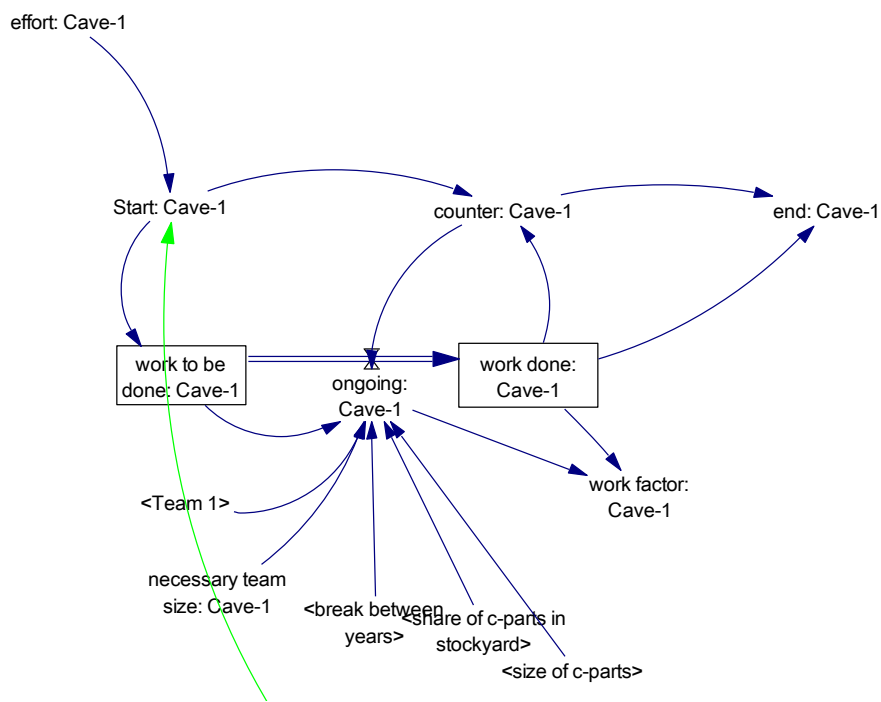


Figure 3-33 Zoom into the modeling construct for the construction phase of the Cave 1 subsystem on the upper middle of Figure 3-32

Figure 3-34 shows the model constructs for the stockyard. The lower left parts represent the delivery dates of the particular deliveries for the construction of the subsystems. These are then bundled to one variable for each subsystem and the c-parts. The variables are used as input for the stock-flow concepts modeling the usage of the stockyard by the subsystems (at the right) and the c-parts (upper left). Each modeling construct for the usage of the stockyard by the subsystems and c-parts is structured similarly. The two outer stocks can be seen as a source and a sink while the stock in the middle represents the *usage of stockyard [by subsystem/c-parts]*.

It increases when the particular parts are delivery and decreases during the particular construction phases as a function of *part-size [of subsystem parts]* and *effort [subsystem]*.

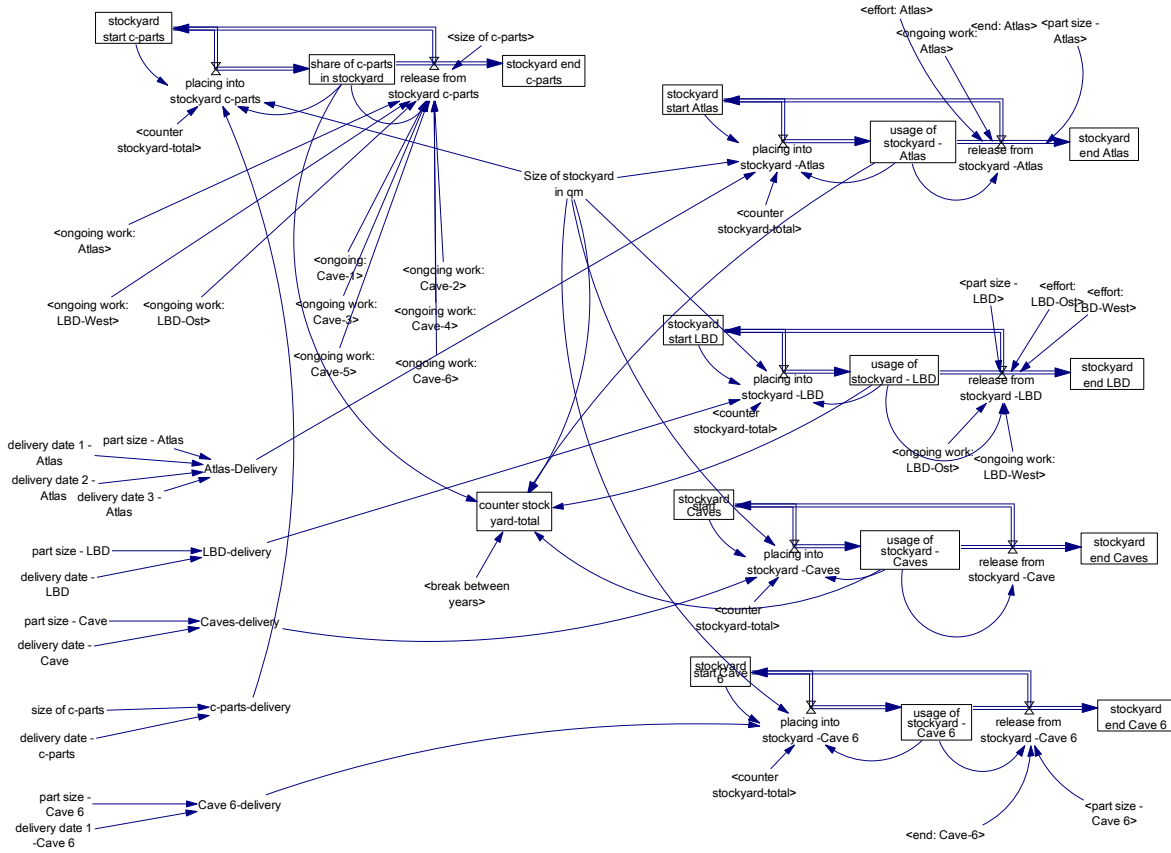


Figure 3-34 System Dynamics model: Model constructs for the stockyard

Following the modeling construct for the construction phases and the stockyard, Figure 3-35 shows the model constructs for the organizational units (right part) and the panel with the simulated process parameters (left part). At the right part the stock allows to analyze when which team is occupied with work. At the left side four overall process simulation parameters are calculated:

- when the construction process is finished (*process completed*)
- the usage of the stockyard over time (*usage of stockyard*)
- how much planned effort is completed (*total effort*)
- how much work has already been done (*total work done*).

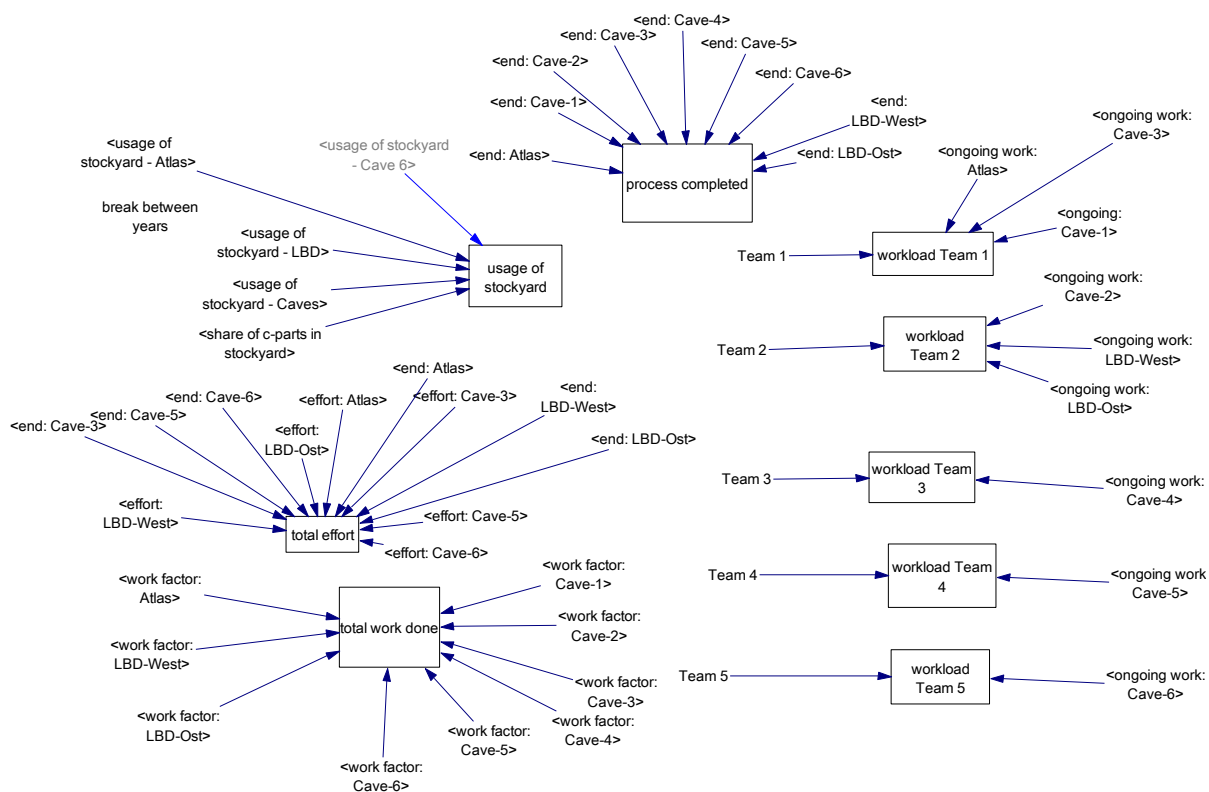


Figure 3-35 System Dynamics model: Panel with simulated process parameters and modeling constructs to represent the organizational units

Quantitative System Dynamics model

The qualitative model was quantified based on the information from the quantitative information acquisition which included the estimated duration and efforts of construction phases and team sizes.

Testing of System Dynamics model

The System Dynamics model was developed in close cooperation with the project management to ensure a sufficient representation of reality by the model. Therefore the System Dynamics model was continuously discussed with them during its development. The simulation results of the quantified System Dynamics model were scrutinized for their realism based on the model tests suggested by Sterman (2000, p. 859). Special focus was directed to: boundary adequacy, structure assessment, parameter assessment, extreme conditions, integration error, behavior reproduction and anomaly as well as sensitivity analysis.

Simulation of different process designs

Based on the tested System Dynamics model, different potential process designs were simulated. The purpose of the simulation was to better understand the process and its relations and thereby identify previously unknown risks as well as assessing the behavior of different process designs towards the already known risk of overrunning the schedule of the construction process.

With respect to the previously identified boundary conditions, ten process design alternatives were identified and simulated. Based on the gathered information two major factors for the design of the process sequence could be identified:

- the start of the construction with LBD-West or LBD-East
- the position of the construction phase of Cave 6.

Cave 6 had no precedence relations to the other construction phases and could float freely within the time span of the overall construction. For the analysis five different starting positions of the construction phase of Cave 6, covering the full possible range, were simulated.

Figure 3-36 gives an overview of the sequences of the simulated design alternatives. The left side illustrates the sequences with LBD-West as starting phase and the floating Cave 6. The right side illustrates the sequences with LBD-East as starting phase and the floating Cave 6. The conducted simulations were named West.0x respectively East.0x, indicating the starting phase and the position of Cave 6 by x between 1 and 5. The ending .01 indicates the start of the construction of Cave 6 at 0 weeks, .02 at 12.5 weeks, .03 at 27 weeks, .04 at 37.5 weeks and .05 at 50 weeks.

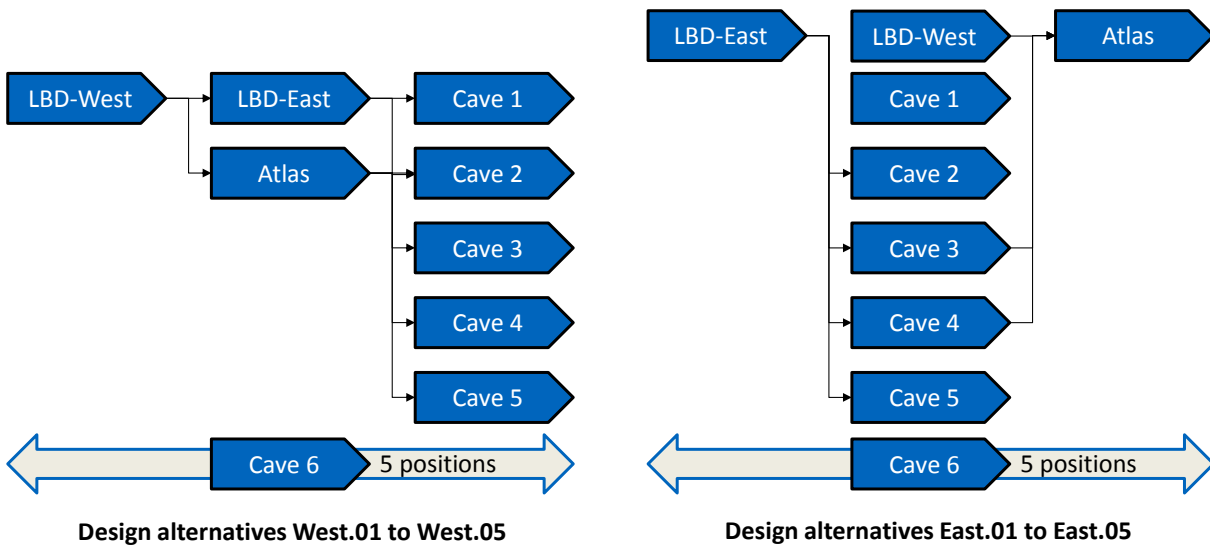


Figure 3-36 Overview of sequences of simulated design alternatives

While at the beginning of the case study the purpose of the simulation was only to optimize the process sequence with respect to its duration, the capacity utilization of the stockyard more and more came to the fore. Initially, it was not thought about the utilization of the stockyard by the project management, but as it came up during discussions in the model development it was included in the model. First results showed that the simulated capacity utilization was close to the maximum space of 180m² which led to a deeper investigation of this factor.

Figure 3-37 and Figure 3-38 show the simulated usage of the stockyard for the ten simulated design alternatives over time. The red line indicates the maximum capacity of the stockyard of 180m². Due to the highly uncertain environment a control limit at 150m² was introduced, which

should not be exceeded by the simulation. Figure 3-37 shows the simulation results for the West.0x design alternatives and Figure 3-38 the results for the East.0x alternatives.

The results of the West.0x alternatives in Figure 3-37 show that the maximum capacity of the stockyard is not reached by any of the alternatives. However, West.01 and 02, slightly cross the control limit of 150m². For all alternatives the project duration is 77 weeks (not directly indicated in Figure 3-37). Two early peaks can be seen which are sourced by the deliveries of the Atlas parts. It can be seen that the maximum usage of the stockyard is lower for the design alternatives West.03 to .05. The block of higher usage of the stockyard which is moving throughout the simulations of the design alternatives is caused by the start of the construction of cave 6. The alternative West.05, even though not delaying the project finish, finishes very close to the overall project finish.

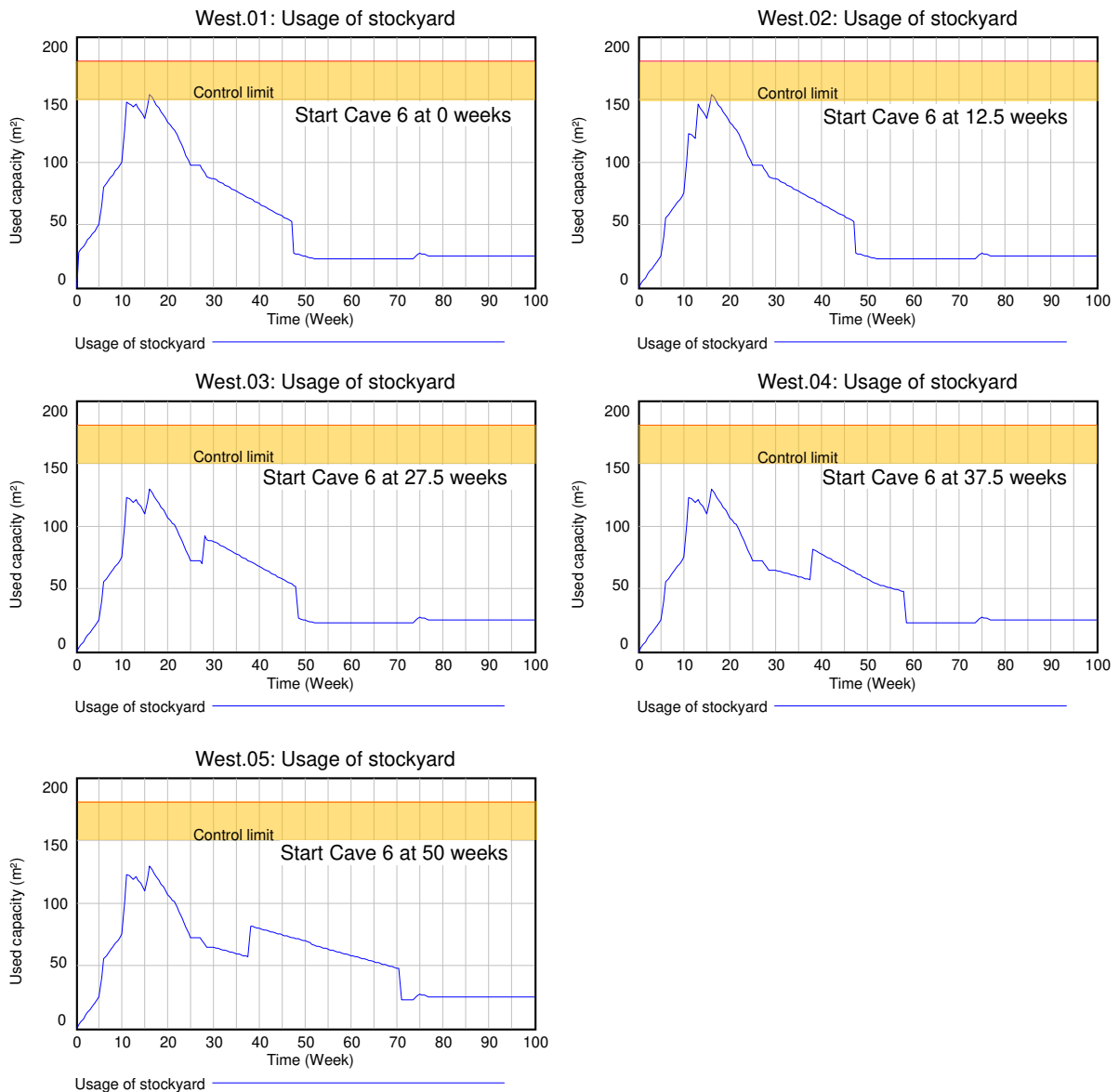


Figure 3-37 Simulated usage of the stockyard for West.01 to .05

The results of the East.0x design alternatives in Figure 3-38 show that also these alternatives do not reach the maximum capacity of the stockyard, but exceed the control limit. For all alternatives the project duration is 86 weeks. While all five design alternatives that start with LBD-West do not significantly have more than 150m² usage of the stockyard, all LBD-East design alternatives do cross this usage value. The alternatives 03. to .05 show a flattened peak compared to East.01 and .02. East.03 shows a double peak constellation due to the start of cave 6 directly after the break between the turn of the years in week 25 and 26. The block of higher usage of the stockyard which can be especially seen in the simulation of the design alternatives East.04 and .05 and is again caused by the start of the construction of cave 6. Similarly to West.05, cave 6 of East.05 finishes very close the overall project finish, thus delays of cave 6 lead to an overall project delay.

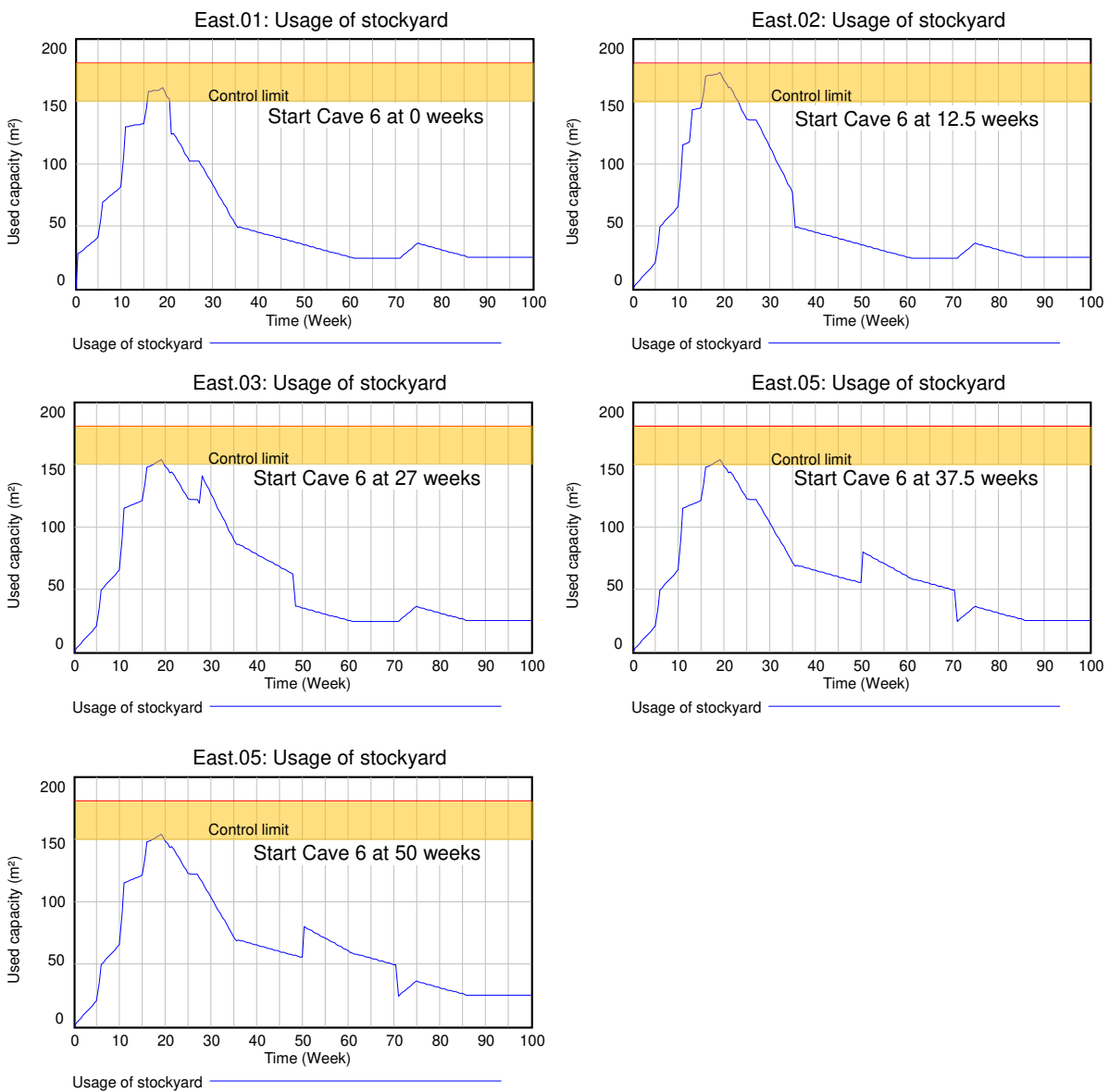


Figure 3-38

Simulated usage of the stockyard for East.01 to .05

Comparing the simulation results of all ten design alternatives West.03 and West.04 were identified as most promising. West.01 and West.02 showed inferior results as it was desired to have a constant usage of the stockyard without major peaks. West.05 and East.05 were not further considered as the finish of cave 6 was so close to the overall project finish that any unplanned delays of the construction could lead to an overall project delay, thus comprised more inherent risk of project delay than the other alternatives. East.01 to East.04 reached the control limit of used capacity, thus are not further considered as better alternatives were found.

Scenario analysis of West.03 and West.04

Based on the previous simulation results West.03 and West.04 were chosen for a deepened scenario analysis. To analyze the robustness of these two design alternatives, the delivery date of the parts of the Atlas subsystem was delayed by 15 weeks and also a delayed and preponed start of construction of cave 6 by 15 weeks each was simulated. The delayed delivery date of the Atlas parts was also combined with the potential deviations of the start of construction of cave 6, leading to 5 scenarios for each design alternative.

To differentiate the simulation runs, the delay of the Atlas delivery is indicated by the supplement of a `_A+15`, the delayed start of the cave 6 construction by a `_C6+15` and its preponed start by `C6-15`.

Figure 3-39 shows the simulated scenarios for West.03. The original simulation run is shown in the middle of the left column. The upper row shows the results for a preponed start of construction of cave 6 while the lower row shows the results for a delayed start. The right column indicates the results with a delayed delivery of the Atlas parts.

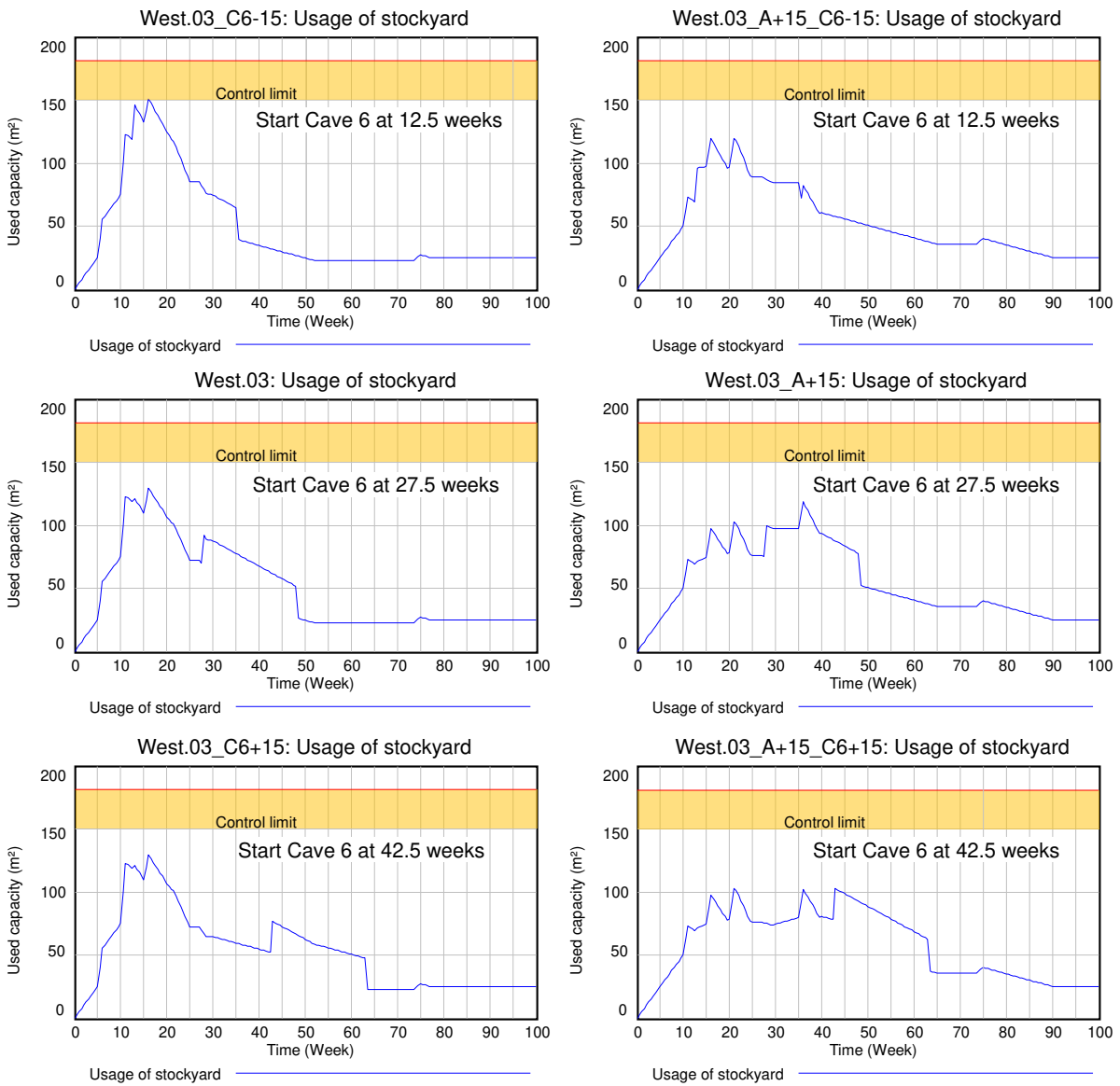


Figure 3-39 Simulated usage scenarios of the stockyard for West.03

Based on the simulation the different usage scenarios of West.03 were analyzed. Table 3-11 shows the analysis results.

Table 3-11 Analysis of the simulated usage scenarios for West.03

<p>West.03_C6-15 Duration: 77 weeks Compared to the original West.03 simulation no delay can be observed. The usage of the stockyard shows an early peak touching the control limit.</p>	<p>West.03_A+15_C6-15 Duration: 90 weeks Compared to the original West.03 simulation a delay can be observed. The stockyard is used relatively steady at the beginning and its usage decreases steadily around week 36 matching the course of West.03_A+15 around week 50</p>
<p>West.03 Duration: 77 weeks Original simulation run already shown in Figure 3-37. An early peak of stockyard usage can be seen as well as a second block of increased usage starting at week 27.5 caused by the start of construction of cave 6.</p>	<p>West.03_A+15 Duration: 90 weeks Compared to the original West.03 simulation a delay can be observed. The stockyard is used relatively steady at the beginning and its usage decreases sharply around week 47</p>
<p>West.03_C6+15 Duration: 77 weeks Compared to the original West.03 simulation no delay can be observed. The usage of the stockyard shows the identical early peak. The 15 weeks delay of increased usage of the stockyard can be clearly seen, but does not influence the overall duration yet.</p>	<p>West.03_A+15_C6+15 Duration: 90 weeks Compared to the original West.03 simulation a delay can be observed. The stockyard is used relatively steady at the lowest level compared to the other simulation runs at the beginning and its usage matches the course of West.03_A+15 around week 64.</p>

Figure 3-40 shows the simulated scenarios for West.04.

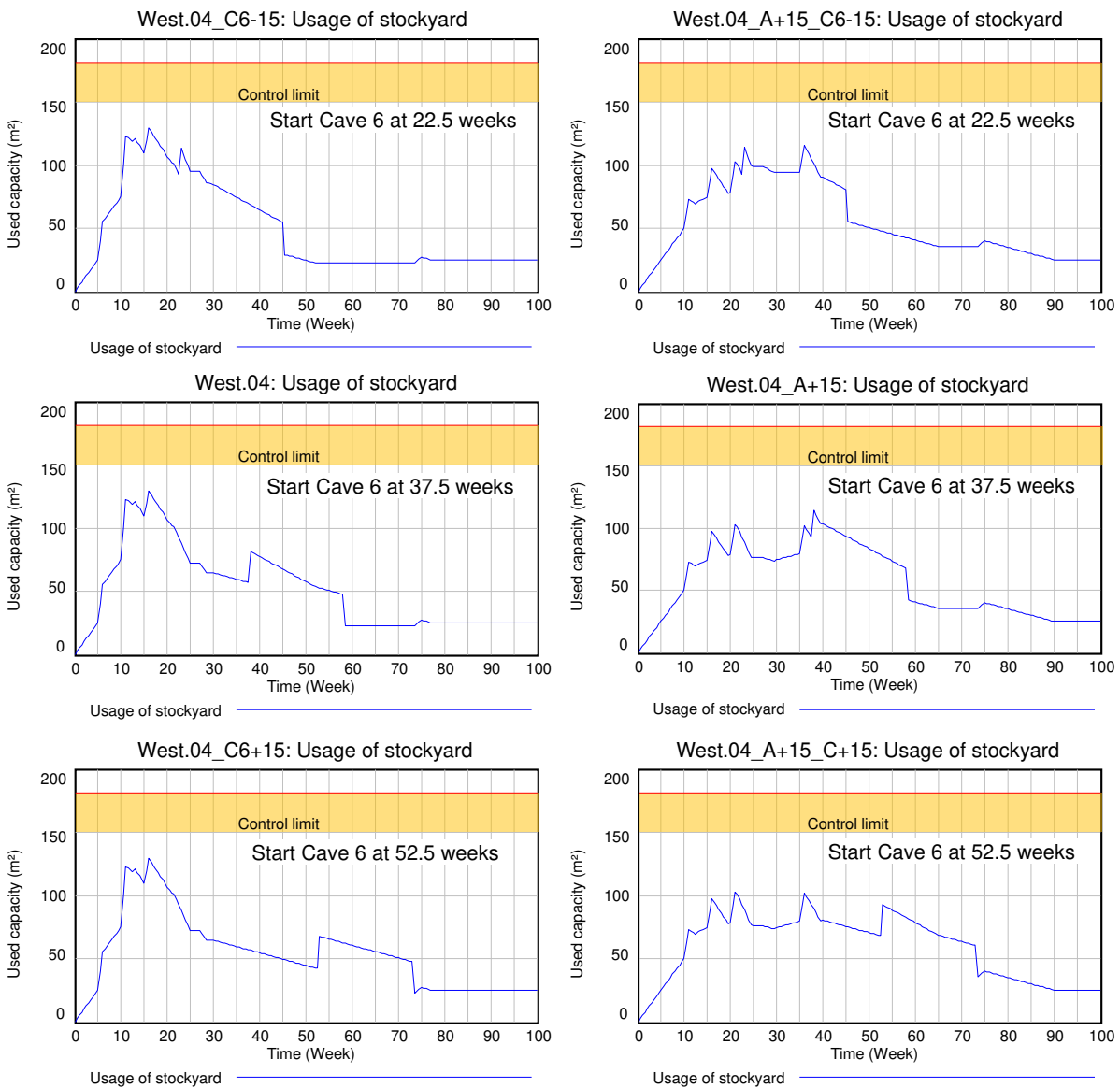


Figure 3-40 Simulated usage scenarios of the stockyard for West.04

Based on the simulation the different usage scenarios of West.03 were analyzed. Table 3-12 shows the analysis results.

Table 3-12 Analysis of the simulated usage scenarios for West.03

<p>West.04_C6-15 Duration: 77 weeks Compared to the original West.04 simulation no delay can be observed. The usage of the stockyard shows an early peak as West.04 but decreases more steadily reaching a low level of constant usage around week 45.</p>	<p>West.04_A+15_C6-15 Duration: 90 weeks Compared to the original West.04 simulation a delay can be observed. The usage of the stockyard at the beginning shows a lower peak than the original simulation, but therefore a longer period of high usage. This period decreases sharply at week 45 and then relatively steady until week 90. It matches the course of West.04_A+15 around week 57.</p>
<p>West.04 Duration: 77 weeks Original simulation run already shown in Figure 3-37. An early peak of stockyard usage can be seen as well as a second block of increased usage starting at week 37.5 caused by the start of construction of cave 6.</p>	<p>West.04_A+15 Duration: 90 weeks Compared to the original West.04 simulation a delay can be observed. The stockyard is used relatively steady at a high level at the beginning but not reaching the usage values of West.04. The usage decreases sharply around week 57.</p>
<p>West.04_C6+15 Duration: 77 weeks Compared to the original West.04 simulation no delay can be observed. The usage of the stockyard shows the identical early peak. The 15 weeks delay of increased usage of the stockyard can be clearly seen, but does not influence the overall duration yet.</p>	<p>West.04_A+15_C6+15 Duration: 90 weeks Compared to the original West.04 simulation a delay can be observed. The stockyard is used relatively steady at the lowest level compared to the other simulation runs over a long period until week 77. It matches the course of West.04_A+15 around week 77.</p>

The results of Table 3-11 and Table 3-12 show that the process durations of the original simulations West.03 and West.04 change to the same degree for the analyzed scenarios. With exception of West.03_C6-15 the usage of the stockyard is relatively stretched for all scenarios. However scenarios were identified in which the maximum of stockyard usage is reached relatively late for West.03 as well as West.04 (i.e. West.03_A+15_C6+15, West.04_A+15). A late maximum of necessary stockyard capacity was not desirable: As the stockyard is empty at the beginning, this space may be misused by others for storing or operation. If the stockyard seems to be disproportionately large and unused for a long period this likelihood increases, may leading to a stockyard overflow when the capacity is actually desired. The necessary stockyard capacity over time decreases faster for West.03 than for West.04.

For West.03 the start of construction of cave 6 should not be preponed (West.03_C6-15) as the stockyard capacities for the parts of the Atlas subsystem and cave 6 were necessary at the same time, leading to a strong peak. However, this peak is at the beginning of the construction period where misuse of stockyard capacity is not likely.

Additionally for the original West.03 scenario the start of construction of cave 6 is at week 27 which is directly after the break at the change of the years, which is expected to have a positive psychological effect as all previous work must be completed at the old year, while the new phase of construction of cave 6 can directly start after the break. Even though for significant delays or preponing of phases the usage of this effect would not be feasible, this option did not exist at all for West.04.

After considering all aspects, it was concluded that West.03 and West.04 show a behavior for the analyzed scenarios which is similar to a large degree and both led to positive predictions of the overall process. After discussion with the responsible project managers they decided to use West.03 as planned process sequence due to the earlier peaks in necessary stockyard capacity and the chance of using the break at the change of the years as additional motivation effect at the construction site.

3.5.6 Stage 5: Disseminate

In a follow up conversation after the case study its approach and results were discussed with the two responsible project managers from a post perspective and feedback was given. The researchers a priori defined the question given in Table 3-13 which were answered by the project managers for structured feedback.

Table 3-13 Questions of the evaluation with regard to the results of the case study

No	Question
Q1	Do you think that the MDM was useful to reveal the dependencies of the process and improve their understanding?
Q2	Do you think that your understanding of the dependencies and risks of the process has improved because of the System Dynamics Simulation?
Q3	Has your understanding regarding the influencing factors and critical elements changed?
Q4	Did you expect further knowledge which has not been provided by the System Dynamics Simulation?
Q5	Do you think that the results of the System Dynamics Simulation could be useful to optimize the process?
Q6	Do you think that the combined use of matrix-based and System Dynamics approaches is appropriate for optimizing the CALA construction process?

While one project manager thought the MDM was useful to reveal and understand the dependencies of the construction process, the other manager would have preferred a graph visualization more (Q1). This aligns with theory that visualization preferences individually differ (Lindemann et al., 2009).

Both project managers agreed that the simulation created added value in understanding the process and its risks (Q2). A duration estimate could be determined, thus insights on a known risk could be gained. Additionally, the capacity of the stockyard was identified as previously unknown risk. By investigating different sequences, process configurations could be identified which show a decreased risk of stockyard overflow (West.03 and West.04). The simulated scenarios with deviations of West.03 and West.04 additionally showed their robustness and led to the choice of West.03 as process sequence. The project managers explicitly mentioned that the simulation encouraged to think about such challenges at all.

During the continuous operation of the approach new questions consistently emerged. While the general effects of influencing factors were previously known, the simulation allowed for a deepened analysis of single risks (Q3). For example the laser system could be identified as very critical element and it is now thought that with the taken actions the risk of stockyard overflow is justifiable.

One of the project managers would have desired a more detailed risk analysis on a lower level of abstraction (Q4). Unfortunately the wish could not be satisfied as it was expressed quite late and would have corresponded with significant additional effort.

The same project manager did not like the word “optimize” in Q5. He argued that a process could only be optimized if it is well known. As the process analyzed by the case study is a construction process it comprises a lot of uncertainties and can therefore not be optimized. However, he strongly agreed that the System Dynamics model was useful for planning the process. The other project manager agreed on the original question (Q5). Based on the results of the case study a request for proposal was sent to the manufacturer of the LBD to build up the system from the west side which was initially planned to be built from the east side. Additionally a more concrete logistics concept was defined based on the assumptions of the simulations. Both managers would have liked a “at the push of a button” multi-run simulation environment which automatically optimizes the process according to a command variable. Both could imagine a predefined simulation which could be used for different construction sites. It should be a rather abstract simulation which can be quickly customized for each site with the available data. However they noted that the acceptance of the simulation results could be difficult due to the underlying black box principle on the given abstraction level. Both project managers agreed on the appropriateness of the combined use of matrix-based and System Dynamics approaches for the problem of the case study for process planning and risk analysis (Q6).

3.5.7 Implications of the Case Study

Provision of heuristics for quantification

Considering all case studies, the step from the qualitative System Dynamics model towards a quantified model which can be simulated seems to be of great importance. Depending on the chosen qualitative modeling constructs (i.e. the type of rework cycle or stock-flow construct used) and the boundary conditions of the model, the necessary quantitative information differs. This step of incorporating quantitative information in the model needs to be supported in at least two ways:

- Which quantitative information is necessary for the model?
- How can this quantitative information be derived?

Both question could be supported by heuristics for quantification. Some of these heuristics may be intuitively obvious, but they must be made explicit if they are to be later used in an automated process (Paulson & Wand, 1992). Browning & Yassine (2010) state that heuristics are especially important for:

- very large problems
- constructing initial solutions
- increasing the speed and simplicity of modeling
- decreasing the entry barrier in modeling.

The heuristics shall give guidelines from a modeler's point of view. In the literature, various heuristics as well as criteria for credible heuristics can already be found (Biemans et al., 2001). These and new heuristics should be documented in an accessible way to support the modeling process.

Differentiation of System Dynamics modeling constructs by abstraction level

Furthermore a need for differentiation was identified. While for the case studies 1, 2 and 3 rework cycle concepts were used as System Dynamics modeling constructs, simpler stock-flow concepts were used in this case study. Additionally to the implication of case study 3 that guidance is necessary when to use which rework cycle concept, guidance is necessary when to use rework cycles at all and when other modeling constructs may fit better.

Exposure to parameter uncertainty

A significant variation in output can be caused by parameter uncertainty (Rahmandad & Sterman, 2008). This was especially spotlighted by the scenario analysis in this case study. However, uncertainty is a significant driver in the context of engineering design process.

Process parameters need to be estimated, and estimation errors and uncertainties are unavoidable. Because of the existence of this uncertainty, inputs of the simulation may not be well-defined, and so their output values and the evaluating criteria for different process options are vague and difficult to compare with each other. It is necessary to reduce the effect of this uncertainty. (Liu & Liu, 2010)

3.6 Summary of the case studies

Within the four conducted case studies 13 implications for the framework for structure-based System Dynamics Analysis of Engineering Design Processes were identified. These implications, listed in Table 3-14, were together with a literature analysis taken as a basis to develop the framework which is explained in the following section.

Table 3-14 Identified implications within the case studies

Case study	Implication
1	General steps of the structure-based System Dynamics Analysis framework concept
	Model transformation concept from MDM models to System Dynamics models
	Use of multiple rework cycles within one model
	Structured representation of dependencies with the System Dynamics model to reduce complicatedness
2	Detailing of the framework concept
	Refining the model transformation concept
	Guidance for model testing
3	MDM metamodel for the structure-based System Dynamics Analysis
	Guidance for when to use which rework cycle
	Reproducibility of quantification process (Quantification sheets)
4	Provision of heuristics for quantification
	Differentiation of System Dynamics modeling constructs by abstraction level
	Exposure to parameter uncertainty

Figure 3-41 gives an overview of the conducted case studies and the literature analysis. The Figure especially highlights the combined approach of this thesis to derive the framework for structure-based System Dynamics Analysis. The implications from each case study, given in Table 3-14, were directly introduced into the framework as well as implications were detailed and refined within further case studies. All case studies were accompanied by a literature analysis whose results were also introduced into the case studies as well as the framework and are mainly documented in section 2.5.

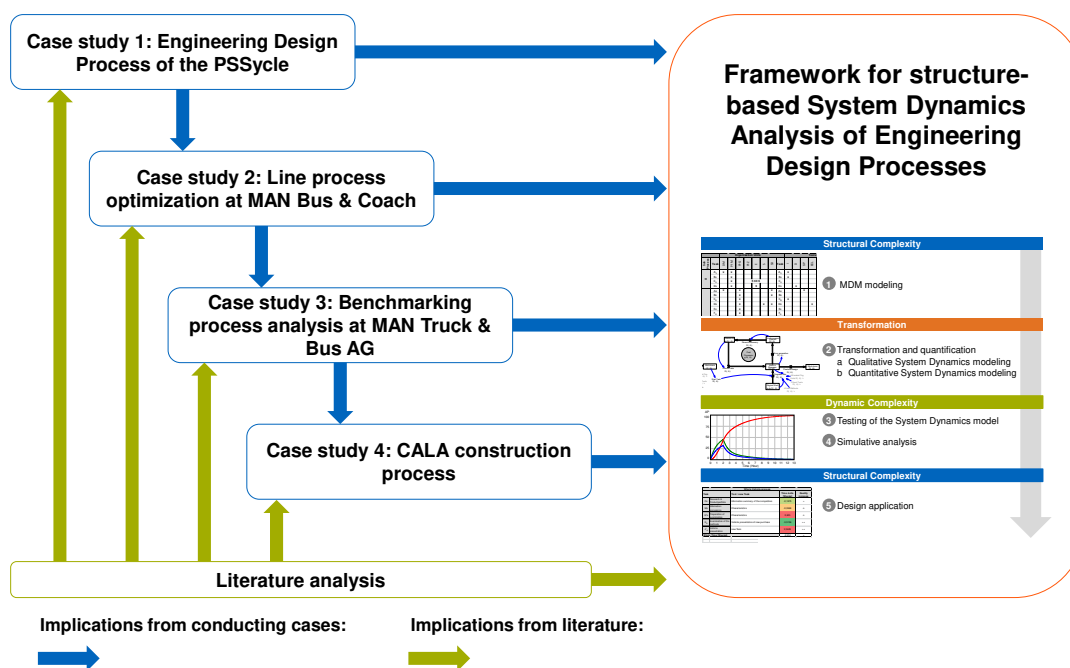


Figure 3-41 Overview of conducted case studies and literature analysis indicating their contribution to the framework

4. Framework for structure-based System Dynamics Analysis of Engineering Design Processes

This chapter presents a systematic methodological framework for the structure-based System Dynamics Analysis of Engineering Design Processes as Prescriptive Study. It has been developed based on the theoretical background, state of the art and conducting case studies from both academic and industrial application. The first section gives a summary of the framework. Subsequently, the procedure of the framework is explained in detail.

The structure-based System Dynamics Analysis presents a systematic, generic methodology that can be applied to engineering design processes. It enables the creation of a conceptual model space for the joint consideration of the dimensions of structural and dynamic complexity for supporting the early phase of engineering design processes. This model space can be used for experimentation to support the early phase of engineering design processes.

4.1 Summary of the framework

The framework for structure-based System Dynamics Analysis of Engineering Design Processes has been developed based on the insights from the identified literature and the conducted case studies. During each of the case studies, additional elements for the framework were identified. Additionally to the framework, tools and methods for the particular steps of the framework are suggested. Referring to the static and dynamic views on engineering design processes as illustrated in Figure 1-1, the procedure of the framework is given in Figure 4-1. The accompanying tools and methods are shown in Figure 4-2.

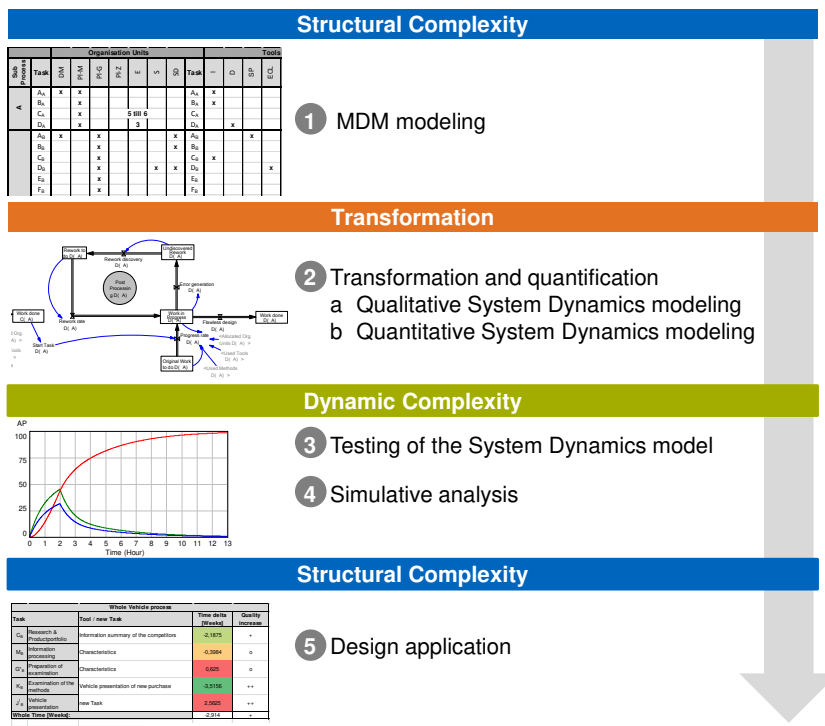


Figure 4-1 Procedure of the framework for structure-based System Dynamics Analysis

The framework combines the views on engineering design processes from the structural and dynamic complexity perspective and offers special support for the transformation between these views.

The starting point for the structure-based System Dynamics Analysis is the **MDM modeling**. Therefore, the first two steps of system definition and information acquisition of the structural complexity management methodology from Lindemann et al. (2009) can be applied. While the original procedure suggests the free definition of domains based on the particular problem description, the structure-based System Dynamics Analysis provides a system definition guideline in form of a predefined MDM metamodel. The metamodel comprises the most common domains and interdependencies in the context of engineering design processes and allows an eased transformation of the MDM into a qualitative System Dynamics model within the subsequent step of the framework. The data gathering for constructing the detailed MDM is supported. A detailed description of the MDM modeling is given in section 4.2.

Based on the proposed **transformation and quantification** concept the MDM is transformed into a System Dynamics model. The transformation concept includes a detailed description of the transformation from the MDM, representing the structural complexity, to a customized System Dynamics model representing the dynamic complexity. Differentiating between the phases of qualitative and quantitative modeling, the transformation concept comprises six tools and methods especially developed to support the transformation. System Dynamics composition panels allow for an eased transformation and management of the structural relations within the System Dynamics model. The System Dynamics classification of engineering design processes for choosing System Dynamics modeling constructs supports users of the framework in finding the correct abstraction level for System Dynamics modeling. Based on the chosen abstraction level, the framework proposes the use of the modeling construct of rework cycles to represent the iterative nature of engineering design processes. As various forms of rework cycles have been developed for different purposes within the past 35 years, a literature-based guideline for the adaptation of rework cycles aims to support in choosing and adapting the most suitable rework cycles for the particular challenge at hand. The result of the transformation is a qualitative System Dynamics model derived from the originating MDM. The transformation is further detailed in section 4.3.1 and followed by the quantification.

During the quantification the qualitative System Dynamics model is populated by equations and values for each component of the model. Information from the MDM not yet incorporated in the model as well as additional information acquired through interviews and available process documentation is used as input. To support this step a set of heuristics to quantify the System Dynamics models of engineering design processes is given. The heuristics thereby base on previous studies from literature as well as insights of this thesis and commonly accepted rules of thumb. As uncertainty is a significant driver in the context of engineering design processes, approaches to deal with parameter uncertainty for example by multi-run simulations with distributed variables are introduced. While the step of quantification of System Dynamics models is known to be crucial but often insufficiently documented, specific quantification sheets are proposed to document the assumptions incorporated in the model and their original

source of information. In section 4.3.2, the heuristics are presented as well as methods to deal with uncertainty and the concept of quantification sheets are provided.

These tools and methods increase the reproducibility of the System Dynamics modeling process and are also the basis for the next step of **testing of the System Dynamics model**. To prove its usefulness for the following analysis, it has to be tested if the created System Dynamics model sufficiently enough represents the real engineering design process for the purpose of the analysis. As an extensive description of verification criteria for System Dynamics models is provided by Sterman (2000), this thesis refers to his work for the testing of the structure-based System Dynamics model in section 4.4.

The tested model can be used for the **simulative analysis** of the behavior of the considered engineering design process. Three different types of analysis which can be conducted with the framework for structure-based System Dynamics Analysis are described in section 4.5. The combined consideration of structural and dynamic complexity decreases the perceived complicatedness of the considered engineering design process and thus, increases the understanding of the overall complexity. This increased understanding supports risk management by uncovering knowable unknown unknowns. Additionally, the created simulation environment allows taking a “what if?” perspective and simulating the effects of potential future conditions. Based on the simulation insights, scenarios can be defined and tested. This includes the impact analysis of events triggered by external or internal factors as well as robustness analysis of engineering design processes considering uncertain conditions. Furthermore, the created simulation environment can be used to assess the performance of different structural process designs of the engineering design process. This benchmarking of different process designs is especially interesting for the early planning phase of engineering design processes as benchmarking allows insights into which structural process design performs best under the assumed conditions.

The **design application** makes use of the results of the system analysis in order to provide insights for the considered complexity challenge. This includes the creation of a complexity analysis report to document solutions for a better handling of the initial challenge in section 4.6.

Figure 4-2 gives an overview of the tools and methods of the framework for structure-based System Dynamics Analysis.

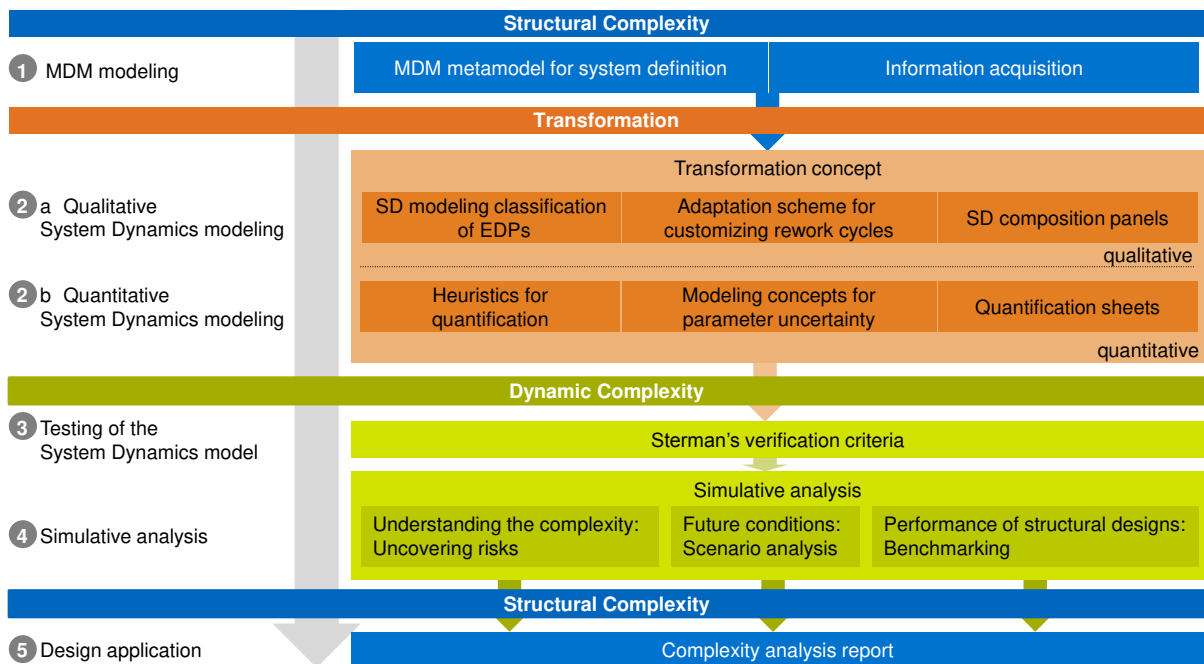


Figure 4-2 Tools and methods of the framework for structure-based System Dynamics Analysis

In the following the five steps of the framework are described including the particular tools and methods.

4.2 Step 1: MDM modeling

The first step of the structure-based System Dynamics Analysis is the MDM modeling. MDMs allow to represent multiple network structures, both within a single domain and across domains, and to capture different relationship types that coexist concurrently. As the MDM is an extension of the DSM and DMM without losing any features of these original structural models, Kreimeyer (2009, p. 106) claims that MDMs are the ideal tool for modeling the structure of design processes. For the framework the MDM is chosen as modeling technique for structural complexity for a number of reasons stated by Kreimeyer (2009):

- The network structure of the process can be modeled in all its facets due to the free choice and adaptability of domains and dependencies.
- Most other existing process models can be translated into MDMs easing the applicability of the framework.
- Qualitative and quantitative models can be combined to some extent, for example weights for nodes or edges can be used or attributes via additional matrices can be introduced.
- Structural characteristics become accessible. This eases understanding the interrelation of structural and dynamic complexity as well as the integration of the structural characteristics into models of dynamic complexity.
- The framework can base on existing work in the field. Kreimeyer (2009) developed the “Meta-MDM with domains and relationship types suited for most modeling and

analysis purposes” which serves as basis for the later presented MDM metamodel of the framework.

For the development of the MDM model, the first three steps of the structural complexity management methodology (Lindemann et al., 2009) can be applied: system definition, information acquisition and deduction of indirect dependencies. However, the developed framework especially emphasizes the first two steps and provides a customized procedure of these steps with respect to its characteristics as these steps are mandatory within the framework. The deduction of indirect dependencies is optional and can be applied as originally described by Lindemann et al. (2009).

MDM metamodel for system definition

A previous version of the MDM metamodel for system definition was already published by the author in Kasperek, Maisenbacher & Maurer (2015).

While the original structural complexity management procedure suggests the free definition of domains based on the particular problem description, for the purpose of the structure-based System Dynamics Analysis a system definition guideline is given.

Within the original structural complexity management procedure the domains are chosen either according to the intended results of the later analysis or according to the existing information sources. With respect to the implications from the case studies a system definition guideline in the form of a MDM metamodel with domains and interdependencies is provided. The metamodel gives orientation when modeling the structural complexity of the engineering design process. It systematizes and collects relevant domains and interdependency types and puts these into a common framework.

The MDM metamodel is based on the “Meta-Multiple-Domain Matrix with domains and relationship types suited for most modeling and analysis purposes” introduced by Kreimeyer (2009). While the domains *event*, *organizational unit*, *resource* and *time* are directly adopted from Kreimeyer (2009), the original *task* domain was renamed into *process step*. With respect to the purpose of the framework, the original domain *artifact* is not considered within this framework. Table 4-1 lists the suggested domains for the MDM modeling step of the structure-based System Dynamics Analysis and a description of each domain, based on the definitions of Kreimeyer (2009).

Table 4-1 List and description of domains of the MDM metamodel based on Kreimeyer (2009)

Domain	Description
Process step	The process step domain collects all entities that describe the execution of work done in the process. Further terms are: tasks, action, activity, unit of behavior and work package.
Event	This domain addresses non-persistent occurrences in time that present a certain action, status or progress. Further terms are: cycle, impact, influence.
Organizational Unit	The organizational unit domain contains all human resources in their respective ordering. Further terms are: staff, responsibility, team, pool, lane, actors, roles, and committee.
Resource	The resource domain is intended for all non-human resources necessary to enable the process execution, such as IT-systems, equipment or knowledge. Further terms are: attribute, mechanism, method, pool, and lane.
Time	The time domain addresses persistent time issues such as the start time of a process step or milestones in the process. Further terms are: attribute, duration, starting time, end time, average time, milestone, and phase.

The various possible interdependency types of Kreimeyer (2009) are reduced for each possible interdependency between the domains. With focus on the joint consideration of structural and dynamic complexity of engineering design processes, only interdependencies that are important for the time dependent process flow were adopted from the original model. As the metamodel can be adapted to different modeling situations, not only one but multiple possible interdependency types are suggested between particular domains. The domains and interdependency types of the MDM metamodel are not compulsory. They should be seen as collection of most common domains and interdependencies, which can and should be adapted for the particular situation at hand. The MDM metamodel is illustrated in Figure 4-3. There, the preferred interdependency type for the framework is indicated in bold letters for each existing combination of domains.

	Process step	Event	Org. Unit	Resource	Time
Process step	<ul style="list-style-type: none"> • precedes temporally 	<ul style="list-style-type: none"> • is influenced by 	<ul style="list-style-type: none"> • is conducted by • is executed by • is required for • supported by 	<ul style="list-style-type: none"> • requires • is supported by 	<ul style="list-style-type: none"> • has duration of • is finished at • starts at
Event			<ul style="list-style-type: none"> • occurs in 	<ul style="list-style-type: none"> • concerns 	<ul style="list-style-type: none"> • happens at
Org. Unit				<ul style="list-style-type: none"> • uses 	<ul style="list-style-type: none"> • is active during • is finished at • starts at
Resource					<ul style="list-style-type: none"> • is available at • is available during
Time					

Figure 4-3

Multiple-Domain Matrix metamodel based on the preliminary work of Kreimeyer (2009)

As shown within the case studies 3 and 4, using the MDM metamodel eases the later transformation of the MDM into a System Dynamics model within the subsequent steps of the framework. However, the domains and interdependencies of the metamodel are only suggestions which can and should be adapted on the particular challenges and data at hand.

Information acquisition

Special support for the system definition phase of the MDM modeling step within the structure-based System Dynamics Analysis framework was developed in form of the MDM metamodel with most common domains and interdependencies. For the phase of information acquisition the framework suggests to use the procedures described by Lindemann et al. (2009). The following paragraphs provide a short summary of the activities of this phase.

Following the system definition by the MDM, information about the dependencies between system elements has to be collected. Information may be acquired from the existing data bases, models, or by interviews. The selected method depends on the availability of information and the specific use case. Interviews require time-consuming workshops with experts; however, if the system in question is only comprised of implicitly known experience knowledge or data that has not yet been documented, interviews provide the only possibility of capturing required information. The main challenge of information acquisition by interviews is to guarantee a high quality of resulting data. (Lindemann et al., 2009)

Lindemann et al. (2009) provide appropriate methods which help to consider all relevant information and to avoid effects resulting from symptoms of fatigue. They define requirements that need to be fulfilled to allow an information acquisition by extraction from available data bases or software tools.

4.3 Step 2: Transformation and quantification

To enable a simulation of the engineering design process behavior with the herein presented framework, the MDM metamodel needs to be transformed into a System Dynamics model. This aligns well with the need for standard System Dynamics structures (Warren, 2014).

Differentiating between the phases of qualitative and quantitative modeling, the transformation and quantification concept comprises six tools and methods especially developed to support this step. Ideas and insights from the transformation framework of Le (2013) were taken as a basis and further developed to derive the transformation concept between MDMs and System Dynamics.

Table 4-2 shows the transformation of the MDM domains into System Dynamics models. While the elements of the process step, organizational unit, and resource domains and their relationships can be directly transformed into elements of a qualitative System Dynamics model, the elements of the event and time domains and their relationships can be used for the quantitative System Dynamics modeling. In doing so the elements and relationships cannot be directly transferred into the System Dynamics model, but serve as a basis for the construction of the necessary differential equations for quantitative System Dynamics modeling.

Table 4-2 Transformation of the MDM domains into System Dynamics models based on Kasperek, Maisenbacher & Maurer (2015)

Elements of MDM domain	Transformed into qualitative System Dynamics model	Used as input for quantification of System Dynamics model	System Dynamics construct used
Process Step	X		stocks or rework cycle constructs
Event		X	
Organization Unit	X		resource pool
Resource	X		resource pool
Time		X	

4.3.1 Step 2a: Qualitative System Dynamics modeling

The result of the previous step of MDM modeling is a MDM of the engineering design process. It comprises the necessary process steps, the available organizational units and resources. The model also comprises events that may occur and affect the engineering design process as well as a time estimate for the particular process steps. As first sub step of the transformation and quantification thus, the bridging the modeling approaches of structural and dynamic complexity, the elements and relations of the MDM need to be translated into a qualitative System Dynamics model. How the elements and relations are transferred depends on the chosen abstraction level. The System dynamics modeling classification of engineering design processes supports this step.

System Dynamics modeling classification of engineering design processes

The classification of rework cycles and the adaption scheme for customizing rework cycles were developed in close cooperation with a master student and are also documented in Produktentwicklung (2014b) respectively Schmidt et al. (2015).

The classification of rework cycles helps understanding the commonalities and distinctions of the System Dynamics models. This is useful because models contain diverse elements and are designed differently to simulate certain behaviors. Both classifications contain various references which help the modeler to identify models with similar abstraction levels and purposes which can serve as a guidance on how to model the engineering design process. Amongst others, existing rework cycles in literature can be classified according to their:

- abstraction level or
- purpose of rework cycles.

Classification by abstraction level

The degree of abstraction determines the scope of the model (Le, 2013). The framework proposes to use the rework cycle concept depending on the level of abstraction of the overall process model. Based on a literature review which can be found in Produktentwicklung (2014b) three levels of abstraction of System Dynamics models were identified: Project, phase and sub-phase level.

Figure 4-4 shows the suggested pyramid which illustrates the use of rework cycle constructs depending on the abstraction level.

The project level is the highest level of abstraction. Le (2013) states that only one rework cycle is necessary to describe the process on the project level. The models of Cooper et al. (2002) and Taylor & Ford (2006) can be allocated to this abstraction level.

The middle of the pyramid in Figure 4-4 shows that the complete project can also be divided into several phases. For example these phases can be design and construction (Parvan et al., 2013) or opportunity selection, preliminary concept definition, full concept definition and product realization (Le, 2013). As these examples show, the phases can be very different in content and amount of work because the optimal breakdown into phases is dependent on what is observed as the complete engineering design process. If an engineering design process is modeled on this phase level, it is represented by a number of rework cycles that usually equals the number of phases. Le (2013) justifies this approach by stating that this ensures that each phase is modeled at the same abstraction level.

The bottom level of the pyramid in Figure 4-4 is the sub-phase level. It is used when one phase of a project is described with several steps or process steps. On the sub-phase level the particular process steps can be modeled as simple stock+flow constructs. They do not have to be transformed into rework cycles as the process steps automatically form rework cycles due to the inherent iterative nature of engineering design processes. This is still incorporated in the underlying sequence of process steps on this level. Taking a more abstract view on engineering design processes, this iterative nature gets lost, thus rework cycle constructs have to be used.

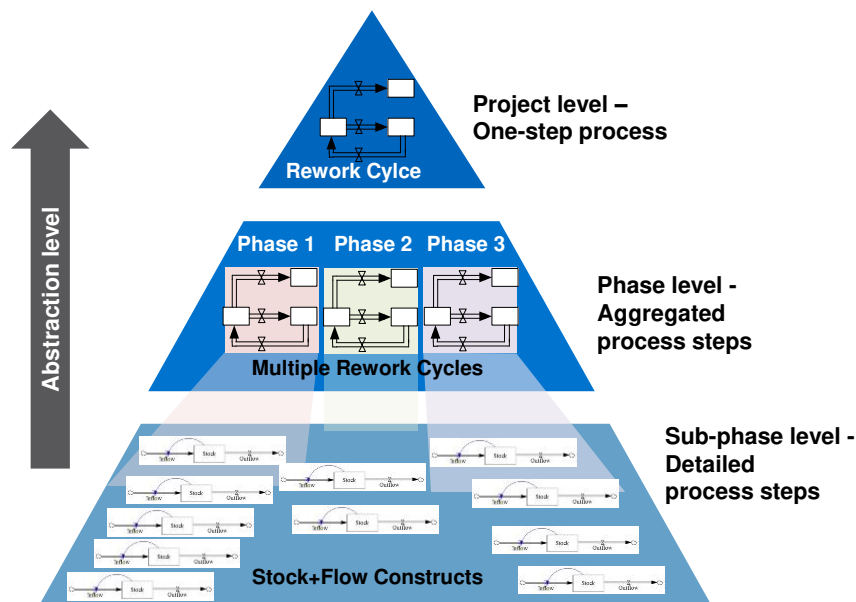


Figure 4-4 Abstraction pyramid illustrating the three levels of abstraction for System Dynamics modeling of engineering design processes – project, phase and sub-phase level – in descending order

When starting to create a System Dynamics model which represents an engineering design process, it is important to decide which degree of abstraction the model should feature. Based on this decision, the size of the model is defined: One rework cycle is necessary to model an engineering design process on the project level and a number of rework cycles up to equaling

the number of process steps within an engineering design process are required for modeling on the phase or on the sub-phase level.

Classification by purpose of rework cycles

Most System Dynamics models in the context of engineering design processes use rework cycles as core modeling construct. The second classification of System Dynamics modeling of engineering design processes is the purpose of these rework cycles. System Dynamics models of rework cycles were developed to conduct studies with various focusses and research objectives. Without intending to be exhaustive, Table 4-3 gives an overview of possible purposes of rework cycles that can be found in the System Dynamics literature. Developers may include structures to capture impacts of project staffing, phase concurrency, testing processes and more. There are rework cycles which pursue more than one of the listed purposes in Table 4-3 and thus, contain several structures that may be incorporated for different purposes.

Table 4-3 Purposes of rework cycles allocated to the references (Schmidt et al., 2015)

Purpose of rework cycle	References
Phase Concurrency	(Ford & Sterman, 1998a, 2003; Kasperek et al., 2014; Le et al., 2012; S. Lee & Sung Lim, 2007; Lin, 2006; Lin et al., 2008; Nasirzadeh et al., 2013; Parvan et al., 2013; Powell et al., 1999; Reichelt & Lyneis, 1999; Ruutu et al., 2011)
Human Factors	(Haslett & Sankaran, 2009; Kasperek et al., 2014; Laverghetta & Brown, 1999; Lisse, 2013; Munoz Hernandez et al., 2013; Reichelt & Lyneis, 1999; Ruutu et al., 2011; Williford & Chang, 1999)
Staffing	(Black & Repenning, 2001; Haslett & Sankaran, 2009; Joglekar & Ford, 2005; Lisse, 2013; Munoz Hernandez et al., 2013; Reichelt & Lyneis, 1999; Repenning, 2000; Taylor & Ford, 2006; Williford & Chang, 1999)
Outsourcing	(Lisse, 2013)
Testing	(Lin et al., 2008; Rahmandad & Hu, 2010)
Tipping Point	(Rahmandad & Al., 2005; Taylor & Ford, 2006)
Cost and Schedule Foresight	(Cooper & Lee, 2009; Lyneis et al., 2001)
Process Improvement	(D'Avino et al., 2005; Repenning & Sterman, 2002)

Various studies exist on the topic of *Phase Concurrency* (Krishnan et al., 1997; Terwiesch & Loch, 1999). Partial concurrent execution of phases reduces the desired time-to-market. The drawback of an increasing overlap between phases, however, is that the product is more prone to change, thus initiating rework in all downstream phases (Powell et al., 1999). Table 4-3 lists authors who included the consideration of phase concurrency in their System Dynamics rework cycle models.

The *Human Factors* mentioned in Table 4-3 take into account that the work rate of employees might not be constant during the engineering design process. The two factors that are most commonly reflected in rework cycles are skill levels of employees (Munoz Hernandez et al., 2013; Ruutu et al., 2011) and effects on the productivity resulting from group effects (Kasperek et al., 2014).

Staffing processes are included in System Dynamics models when the employee turnover of projects is not negligible. Two different methods of capturing changes in the staff level can be

encountered in literature: modeling of hiring and turnover over time (Haslett & Sankaran, 2009; Lisse, 2013; Munoz Hernandez et al., 2013; Reichelt & Lyneis, 1999; Williford & Chang, 1999) and resource allocation over different parts of the project (Black & Repenning, 2001; Joglekar & Ford, 2005; Repenning, 2000; Taylor & Ford, 2006).

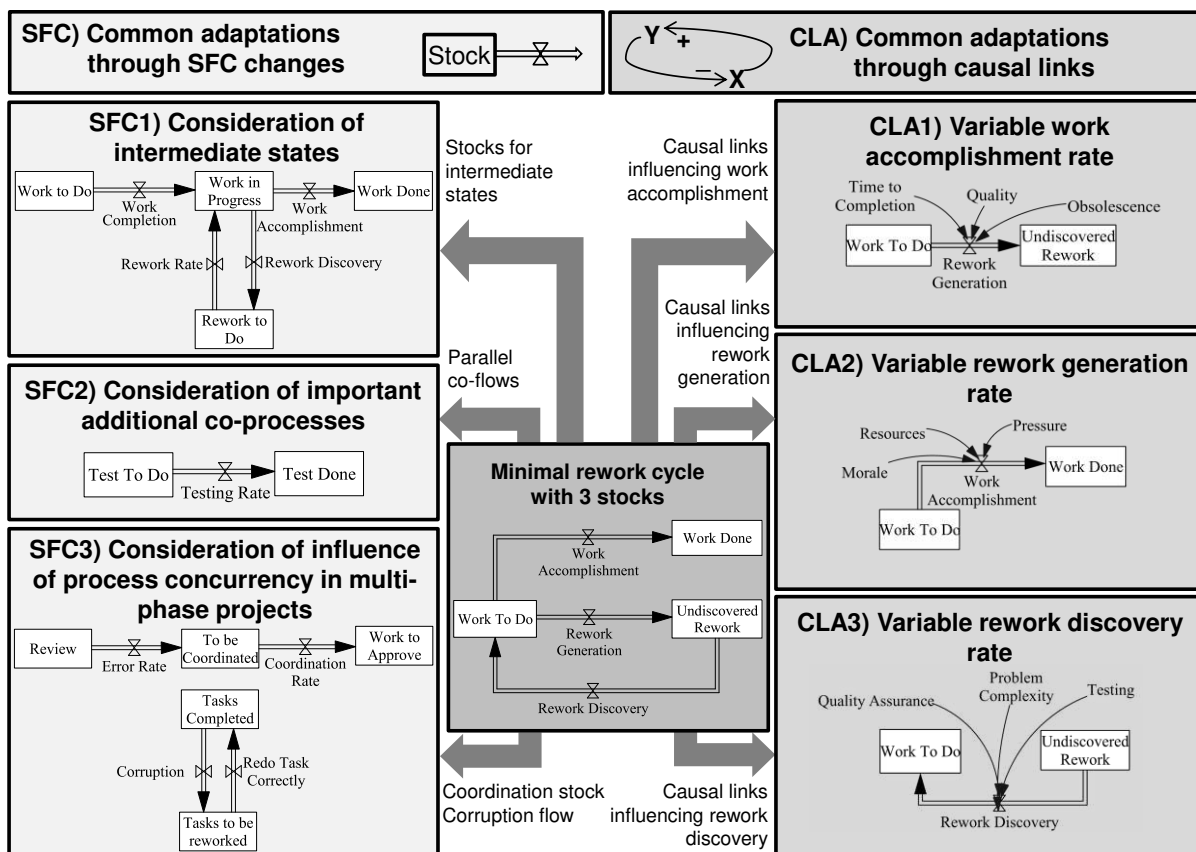
As Table 4-3 shows, features of rework cycles that are less applied are consideration of *Outsourcing and Testing*, as well as the simulation of *Tipping Points*, *Costs*, *Schedules* and measures of *Improvements*.

Modelers need to be aware of which features are to be represented in the rework cycle System Dynamics model. Depending on its purposes, the rework cycle needs to be adapted to enable the simulation of these features. The next supporting tool of the framework, the adaption scheme for customizing rework cycles, provides an overview of the different ways how to adapt rework cycles.

Adaption scheme for customizing rework cycles

The adaption scheme for customizing rework cycles was developed through literature research of 25 different System Dynamics rework cycle models. The rework cycles were analyzed in regard to how they simulate certain characteristics of the engineering design process and how they differ from the simplest version of a rework cycle.

The adaption scheme presents different structures that previous modelers used to technically implement certain behaviors of rework cycles to pursue the defined research objectives. These structures are referred to as adaptations. The adaptations are added to a basic rework cycle model in order to generate a model that simulates the considered engineering design process more accurately. The adaptations are summarized in the adaption scheme shown in Figure 4-5.



© 2015 The authors and IOS Press

Figure 4-5 Rework cycle adaption scheme (Schmidt et al., 2015)

The adaptation scheme is built on a basic rework cycle model. In Figure 4-5, this model is located in the middle. It is characterized by its simple structure that contains the minimum number of independent stocks to model processes with rework (Rahmandad & Hu, 2010). Also Kasperek, Lindinger et al. (2014), Lisse (2013), Nasirzadeh et al. (2013) and Parvan et al. (2013) build their models based on this three-stock rework cycle. Moreover, the model is considered to be simple because of its constant rates. The three rates *Work Accomplishment*, *Rework Generation* and *Rework Discovery* are not influenced by any other variables through causal loops.

The simple model in the middle of Figure 4-5 can be expanded to achieve certain characteristics that are defined by the abstraction level, relationships or desired area of application. The expansion of the model is technically implemented either by means of additional stocks and flows changes (SFC) or causal loops adaptations (CLA).

Rework cycles can be expanded by adapting the SFC by means of adding stocks and flows (Figure 4-5, left) in three ways:

- consideration of intermediate states (SFC 1)
- consideration of important additional co-processes (SFC 2)
- consideration of influence of process concurrency in multi-phase projects (SFC 3).

Another means for adapting rework cycle System Dynamics models are causal loop adaptations (Figure 4-5, right). Certain process behaviors can be represented by adding variables and dependencies between the variables. In most cases, additional variables are used to influence the rates of rework cycles. Using the initial model in the middle of Figure 4-5, there are three rates that are affected by the values of other factors:

- variable work accomplishment rate (CLA 1)
- variable rework generation rate (CLA 2)
- variable rework discovery rate (CLA 3).

Along these six ways of adapting system dynamic models, there are also other ways to model certain features which do not fit into one of the presented categories (e.g. in the models of D'Avino et al. (2005) and Repenning (2000)). However, these adaptations only appear in sporadic cases and thus, are not included in the scheme. The stock and flow changes and causal loop adaptations which were assessed as appearing frequently within the examined cases are explained in the following.

SFC1) Consideration of intermediate states

In many rework cycles, more states – described by stocks – are considered than the three basic ones illustrated in Figure 4-5. Along with additional stocks, new flows are also included, which can be useful in modeling different rates between the process steps. The distinction of rework cycles by the number of stocks is also conducted in the work of Le (2013).

An often used intermediate stock is *Work in Progress*. Figure 4-6 shows a rework cycle which includes this stock. As a consequence, the rework rate is different from the original completion rate. In some processes this could create a benefit, since the assumption of a constant rate for both original and rework would be an improper simplification. In this way the inclusion of intermediate states enables developers to adapt their model so that it accurately reflects the considered engineering design processes.

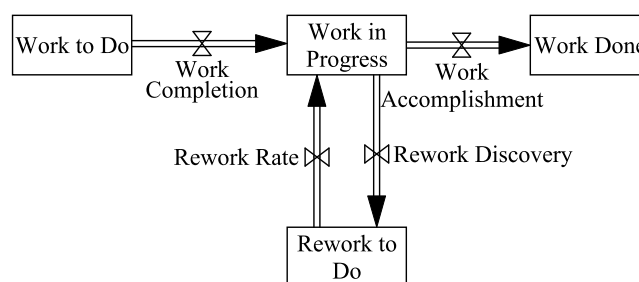


Figure 4-6 Basic model of a rework cycle which is expanded by one stock *Work in Progress*

Table 4-4 lists the references which include intermediate states similar to the one in Figure 4-6. Naming is used differently by the authors in most of the cases. Therefore, Table 4-4 includes a column that lists the names of the additional stocks.

Table 4-4 Intermediate states considered in rework cycles by means of additional stocks allocated to the references (Schmidt et al., 2015)

Intermediate States	References
Tasks Completed not Checked Tasks Approved	(Ford & Sterman, 1998a)
Tasks Completed but not Checked	(Rahmandad & Hu, 2010)
Work in Progress	(Le, 2013)
Tasks to be Reworked	(Lin, 2006; Lin et al., 2008)
Tasks in Testing	(Repenning, 2000)
Work in Quality Assurance	(Joglekar & Ford, 2005)
Quality Assurance Backlog	(Taylor & Ford, 2006)
Known Rework	(Cooper & Lee, 2009; Cooper, 1980; Cooper et al., 2002; Reichelt & Lyneis, 1999)

The modeler of a System Dynamics model is encouraged to rethink the structure of the observed engineering design process and which states need to be captured in the rework cycle. Each crucial state has to be modeled by one stock. The modelers, however, should keep in mind the abstraction level and its relation to the model size.

SFC2) Consideration of important additional co-processes

Some researchers use co-flow structures to consider the dynamics of auxiliary side processes like hiring, training or testing, see Table 4-5. As shown in Figure 4-7, the processes are modeled with co-flows that influence the rework cycle. In Figure 4-7, these processes are staffing and testing. The available staff at a certain point can be calculated as the accumulated difference of the *Hiring* and *Turnover* rate. Tests are processed from the *Test to Do* stock over a *Testing Rate* to the *Test Done* stock. The staff level, for example, may influence the *Work Accomplishment* rate in a way that it increases if more employees are available to conduct the work. In the same way, the number of tests that are done could impact the *Rework Discovery* rate.

Co-processes may be included in the System Dynamics model if the main process is affected and the co-processes show a dynamic behavior during the execution of the main process.

Some developers also integrate co-flows in order to gain additional information during the simulation. In these cases the co-flow does not necessarily affect the rework cycle, such as the change co-flow of Parvan et al. (2013). Their co-flow is included to calculate the resulting costs, but does not impact the rework cycle itself. The inclusion of co-flows can occur on each abstraction level and it enables the developers to pursue specific research objectives.

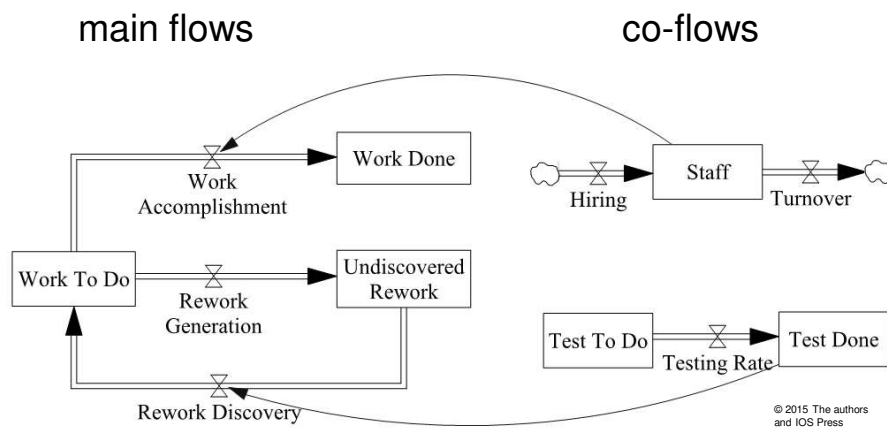


Figure 4-7 System Dynamics models of a rework cycle and co-flows for staffing and testing (Schmidt et al., 2015)

Table 4-5 lists various co-processes that are included in System Dynamics models and their references.

Table 4-5 Co-processes considered in rework cycles as parallel co-flows allocated to the references (Schmidt et al., 2015)

Co-processes	References
Testing	(Lin et al., 2008)
Testing, Change Approval	(Rahmandad & Hu, 2010)
Error Rectification	(Lin, 2006)
Phase Task Change	(Laverghetta & Brown, 1999)
Hiring	(Haslett & Sankaran, 2009; Reichelt & Lyneis, 1999)
Hiring & Training	(Munoz Hernandez et al., 2013; Williford & Chang, 1999)
Change Generation	(Parvan et al., 2013)
Change Generation & Discovery	(S. Lee & Sung Lim, 2007)
Hiring & Training, Completed Work, Expended Effort	(Lisse, 2013)
Error Generation, Hiring & Training	(Ruutu et al., 2011)

SFC3) Consideration of influence of process concurrency in multi-phase projects

Another important SFC adaption is the consideration of the effects of process concurrency. This feature often appears in models of multi-phase projects. The impact of iteration due to releasing flawed work to subsequent phases is considered in these models. This inter-phase relationship can also be called coordination.

Ford & Sterman (1998a) as well as Laverghetta & Brown (1999) model coordination by including a *coordination* stock in their rework cycle. In contrast to the additional stocks for intermediate states introduced above, the *coordination* stock is not part of the process chain but generates an additional iteration loop.

In the model of Ford & Sterman (1998a), the coordination of flawed tasks creates a second loop. Their coordination stock can be fed through two flows. The first flow represents tasks that

are flawed due to inheritance from upstream phases. The second flow is determined by the downstream quality assurance during which discovers flawed tasks and sends them back for coordination (Ford & Sterman, 1998a).

The additional coordination stock enables flawed tasks from other phases to be accumulated in this stock until they can be coordinated. In the real process this could happen, for example, in meetings with the responsible employees of the involved phases (Ford and Sterman, 1998).

The authors Kasperek, Lindinger et al. (2014), Lee & Lim (2007), Lin et al. (2008) and Nasirzadeh et al. (2013) dispense with the coordination stock and solely use a corruption flow to model the influence of process concurrency. Figure 4-8 shows the rework cycle of Kasperek, Lindinger et al. (2014) which iterates flawed work from the *Accomplished Work* stock to the *Remaining Work* stock via a corruption flow. The rate of the flow is calculated with the variables *Cor FW task* and *Cor BW task* which quantify the rework from other process steps that is either flowing to the particular subsequent (forward corruption, FW) or previous (backward corruption, BW) process step.

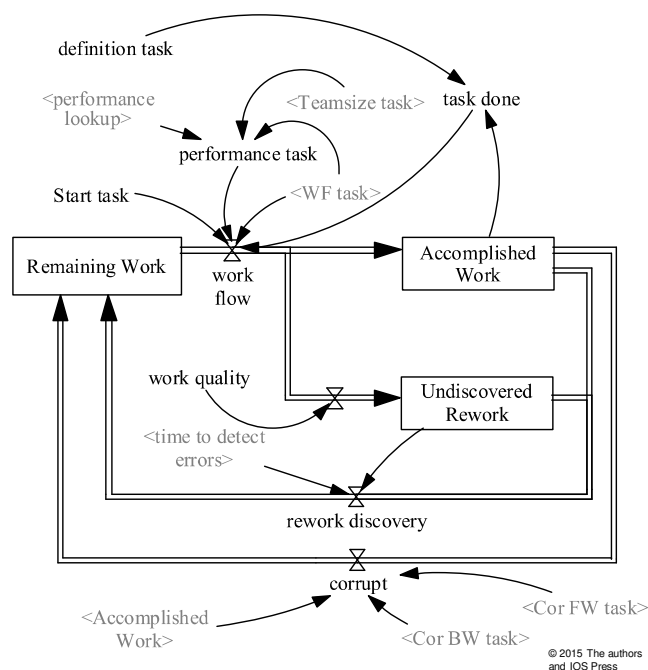


Figure 4-8 Rework cycle with corruption flow (Schmidt et al., 2015)

An overview of existing SFC adaptations for modeling the influence of phase concurrency is provided in Table 4-6.

Table 4-6 SFC implementation of coordination in multi-phase projects (Schmidt et al., 2015)

SFC Adaptations	References
Coordination stock & flow	(Ford & Sterman, 1998a; Laverghetta & Brown, 1999)
Corruption flow	(Kasperek et al., 2014; S. Lee & Sung Lim, 2007; Lin et al., 2008; Nasirzadeh et al., 2013)

For the modeling of concurrent engineering design processes, modelers need to include similar SFC structures to simulate the influence of concurrency.

Next to the adaptation of rework cycles by adding stocks and flows, rework cycles can be also adapted by causal loop adaptations. This means that certain behaviors are represented by adding additional variables and dependencies between the variables. Using the initial model in the middle of Figure 4-5, the three adaptations CLA 1 to 3 which are most important will be described in the following.

CLA1) Variable work accomplishment rate

The work accomplishment during engineering design processes can be influenced by various factors. In System Dynamics models, these influences are modeled by causal loops which affect the work accomplishment rate. For this reason, the adaptation scheme allocates the modeling of variable work accomplishment to the causal loop adaptations. The influencing variables are listed and categorized by the type of modeled influence and the specific factor influencing the accomplishment rate in Table 4-7. Moreover, Table 4-7 includes the authors that capture certain dependencies of the work accomplishment.

Table 4-7 Influences on work accomplishment and associated references (Schmidt et al., 2015)

Category	Factor	References
Company Characteristic	<i>Quality</i>	(Cooper, 1980; Cooper et al., 2002; Ford & Sterman, 1998a; Kasperek et al., 2014; Lin et al., 2008; Lisse, 2013; Reichelt & Lyneis, 1999)
Management Levers	<i>Pressure</i>	(Cooper et al., 2002; Le, 2013; Reichelt & Lyneis, 1999)
	<i>Resources</i>	(Black & Repenning, 2001; Cooper, 1980; Cooper et al., 2002; Joglekar & Ford, 2005; Kasperek et al., 2014; Le, 2013; Lin, 2006; Rahmandad & Hu, 2010; Taylor & Ford, 2006)
	<i>Overtime</i>	(Cooper et al., 2002; Le, 2013; Reichelt & Lyneis, 1999)
	<i>Organizational Changes</i>	(Reichelt & Lyneis, 1999)
	<i>Time to Completion</i>	(Cooper et al., 2002; Lisse, 2013)
Human Factors	<i>Skill Level</i>	(Cooper et al., 2002; Lisse, 2013; Munoz Hernandez et al., 2013)
	<i>Morale</i>	(Cooper et al., 2002; Reichelt & Lyneis, 1999)
	<i>Synergy</i>	(Reichelt & Lyneis, 1999)
	<i>Group Size</i>	(Kasperek et al., 2014; Reichelt & Lyneis, 1999)
Process Factors	<i>Available Work</i>	(Cooper et al., 2002; Ford & Sterman, 1998a; Joglekar & Ford, 2005; Kasperek et al., 2014; Le, 2013; Lin, 2006; Parvan et al., 2013; Ruutu et al., 2011)
	<i>Completed Work in Previous Phase</i>	(Lin et al., 2008)
	<i>Undiscovered Rework in Previous Phase</i>	(Parvan et al., 2013)

The company characteristic *quality* is often a constant factor that defines the percentage of flawless work and thus, the portion of work that moves from the *Work to Do* stock to the *Work Accomplished* stock.

Management levers are, for example, *pressure*, *resources*, *overtime*, *time to completion* and *organizational changes*. The *time to completion* is a factor that arises from the project schedule which is defined by the management. The allocated *resources* directly influence the work accomplishment, whereas *pressure*, *overtime* and *organizational changes* define productivity and therefore indirectly affect the accomplishment of work.

Similarly, the human factors influence the productivity. Different *skill levels* and *morale* are characteristics of individuals that impact work completion. Furthermore, the performance of a team varies with the time due to the development of *synergies* and with *group size*. These factors should be considered in studies that focus on the social impacts on engineering design processes.

Within the category process factors, start conditions and iteration effects are differentiated. Start conditions represent the prerequisites for a phase or process step to start. For example, work completion cannot start unless there is *work available* or the previous phase has not yet produced sufficient output (*completed work in previous phase*). For example, these start conditions are modeled using if-then-else conditions (Kasperek et al., 2014; Lin et al., 2008) or min/max equations (Ford & Sterman, 1998a; Lin et al., 2008). Other factors, such as the amount of *undiscovered rework in a previous phase* continuously affect the work accomplishment throughout the whole process.

CLA2) Variable rework generation rate

In most cases, rework generation within engineering design processes cannot be described with a constant rate. Therefore, many System Dynamics model developers include additional variables to demonstrate the varying behavior of rework generation. Table 4-8 lists various factors influencing the generation of rework, sorted into categories and allocated to authors who included them in their System Dynamics models.

Table 4-8 Influences on rework generation and associated references (Schmidt et al., 2015)

Category	Factor	References
Company Characteristics	<i>Quality</i>	(Cooper, 1980; Cooper et al., 2002; Ford & Sterman, 1998a; Kasperek et al., 2014; Le, 2013; Lin, 2006; Lin et al., 2008; Lisse, 2013; Parvan et al., 2013; Rahmandad & Hu, 2010; Reichelt & Lyneis, 1999)
	<i>Work Completion</i>	(Cooper, 1980; Cooper et al., 2002; Ford & Sterman, 1998a; Kasperek et al., 2014; Lin, 2006; Lin et al., 2008; Lisse, 2013; Parvan et al., 2013; Rahmandad & Hu, 2010; Reichelt & Lyneis, 1999)
	<i>Target Design Maturity</i>	(Le, 2013)
Project Factor	<i>Believed Design Maturity</i>	(Le, 2013)
Management Lever	<i>Time to Completion</i>	(Lisse, 2013)
Iteration Effect	<i>Undiscovered Rework in Previous Phase</i>	(Parvan et al., 2013)
Other	<i>Obsolescence</i>	(Reichelt & Lyneis, 1999)

Table 4-8 shows that rework generation may be dependent on the *work completion* which is among the company characteristics. The reason for this relationship is that the rework generation rate is often calculated out of the work completion or accomplishment rate and the *quality*. The third company characteristic is *target design maturity*. Le (2013) uses this variable in order to compare it with the project factor *believed design maturity* in order to trigger iteration in the rework cycle as well as the start of a subsequent phase.

The value of the variable *time to completion* terminates the process in the example of Lisse (2013), and thus the rework generation corresponding to the work accomplishment ends. Parvan et al. (2013) capture the fact that the amount of *undiscovered rework in the previous phase* increases the error generation in the next phase.

Another factor that can initiate rework is *obsolescence*, which indicates that already accomplished work becomes obsolete and thus, worthless (Reichelt & Lyneis, 1999).

CLA3) Variable rework discovery rate

The discovery of errors is usually modeled as being variable. Many authors include dependencies on other variables in their models to achieve a better approximation of the observed process. Factors that can be found in the literature are listed in Table 4-9 and allocated to different categories.

Table 4-9 Influences on rework discovery and associated references

Category	Factor	References
Company Characteristics	<i>Probability of Discovering Errors</i>	(Ford & Sterman, 1998a; Lin et al., 2008; Rahmandad & Hu, 2010)
	<i>Undiscovered Rework</i>	(Kasperek et al., 2014; Lin et al., 2008; Parvan et al., 2013)
	<i>Quality</i>	(Ford & Sterman, 1998a; Lin et al., 2008; Rahmandad & Hu, 2010)
	<i>Quality Assurance / Testing</i>	(Ford & Sterman, 1998a; Lin et al., 2008; Nasirzadeh et al., 2013; Rahmandad & Hu, 2010; Taylor & Ford, 2006)
Project Factors	<i>Problem Complexity</i>	(Le, 2013)
	<i>Perceived Progress</i>	(Lisse, 2013; Reichelt & Lyneis, 1999)
Management Lever	<i>Pressure</i>	(Taylor & Ford, 2006)
	<i>Resources</i>	(Joglekar & Ford, 2005)
Iteration Effects	<i>Dependence on Previous Phase</i>	(Ford & Sterman, 1998a; Kasperek et al., 2014; Lin, 2006; Lin et al., 2008)
	<i>Rework Completion</i>	(Lin et al., 2008)
	<i>Quality Assurance / Testing</i>	(Ford & Sterman, 1998a; Lin et al., 2008)
	<i>Progress</i>	(Parvan et al., 2013)
Other	<i>Time</i>	(Kasperek et al., 2014; Le, 2013; Parvan et al., 2013)

Company characteristics are, for example, the *probability of discovering errors* and the amount of *undiscovered rework*. The higher these variables are, the more rework is discovered. The

company might also conduct *test* or *quality assurance* processes which also influence error finding.

The project factor *problem complexity* is used by Le (2013) to describe how fast design maturity builds up and thus, if there is need for rework. The other variable related to the project is *perceived progress*. In the model of Reichelt and Lyneis (1999), this factor continuously influences the rework discovery rate.

The management lever *pressure* is modeled as increasing the fraction of rework and thus, the discovery rate in the rework cycle of Taylor & Ford (2006).

The iteration effects can be divided into two groups: impacts that originate from elements in previous phases and impacts of subsequent phases. Lin et al. (2008) and Kasperek, Lindinger et al. (2014), for instance, describe a multiplicative relation between the rework rate of the previous phase and the corruption rate which is combined with the rework discovery rate of a subsequent phase. In the same equation, they include *dependence* which is a constant factor that reflects the percentage of the task that is affected by the change made in a previous phase. Impacts from subsequent phases can be defined, for example, by the *quality assurance* of subsequent phases (Lin et al., 2008) or the work *progress* in these phases (Parvan et al., 2013).

Another important driver for rework discovery is *time*. Two main strategies can be found in the literature for modeling this influence. Le (2013) triggers a fraction of the rework discovery by scheduling discrete times for review meetings in which it is more likely that the need for rework will be identified. Other authors model the continuous, time-dependent development of rework discovery either with equations (Parvan et al., 2013) or look-up tables (Kasperek et al., 2014).

When building a System Dynamics model, all relevant factors within the engineering design process need to be considered and it has to be decided which of them have to be included in the model. Furthermore, the selected variables need to be allocated to the affected rates. Models may include rates other than the three presented rates - work accomplishment rate, rework generation rate and rework discovery rate - but the procedure stays the same. The previously listed variables serve as suggestions for future models.

The adaption scheme in Figure 4-5 guides the modeler deciding on which stocks, flows, variables and causal loops should be included. After adding these elements to the rework cycle, the structure of the SD model is established.

System Dynamics composition panels

While the two previous tools and methods focused on the abstraction level and particular elements incorporated in the System Dynamics model, the System Dynamics composition panel focusses on the modeling of relations. Facing the large amount of relations that may have to be included in the System Dynamics model when using multiple rework cycles, led to the development of composition panels. This counters the complicatedness of the amount of relations and allows to easily change relations between process steps. The structure of the panel is based on the system representation within structural models.

Depending on the chosen abstraction level and its domain, elements of the MDM are transformed to stock and flow concepts, resource pools or rework cycles. Especially if rework

cycles are chosen, the complexity of the model increases as they allow to model a more complex behavior with more influencing factors, than within the MDM. If all relations between elements and influencing factors are incorporated in the System Dynamics model, its complexity grows rapidly.

The idea behind composition panels is to separate the modeling of relations from the modeling of the system elements. While one part of the model illustrates the elements represented in the model by its System Dynamics modeling constructs, the other part shows the relations between these elements in the so called composition panels. From a technical point of view, the computational concept of pointers (for example referred to as shadow variables in the Vensim® simulation software) can be used.

This separation offers the advantage to clearly breakdown the model. This offers the possibility to easily change relations within the model and simulate the influence on the overall process, while keeping the dynamic modeling core constructs (for example the rework cycles) constant. This supports the accurate and quick modeling of different process configurations.

The complicatedness of the system relations can be further decreased by modeling the relations of the DSM representing the sequence of process steps in two different composition panels. Figure 4-9 illustrates a DSM of the engineering design process sequence and the corresponding composition panels modeling the sequences within System Dynamics. In the Figure, A-F start / done / feedback / rework represent variables, thus elements of the simulation each of them referring to the particular process step A-F.

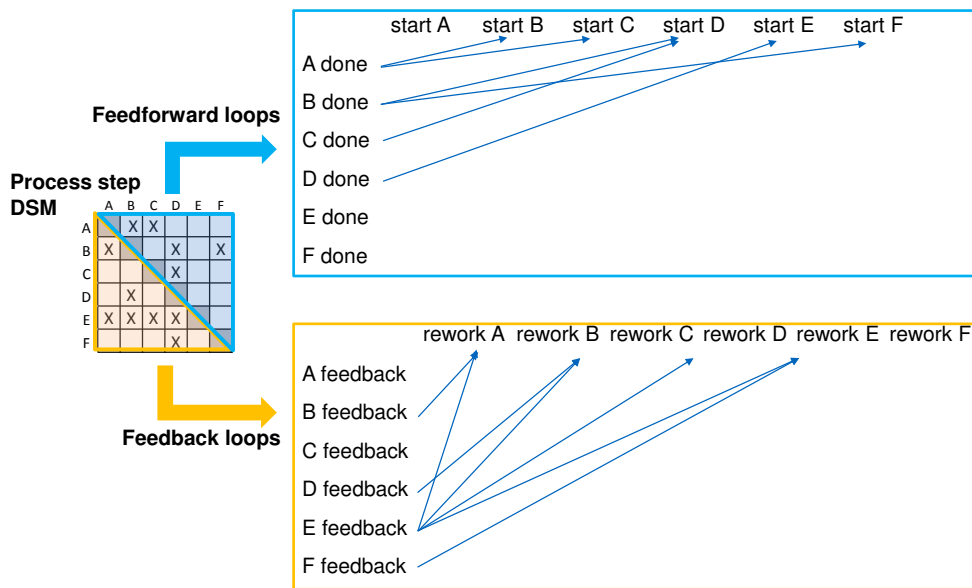


Figure 4-9 DSM-based composition panel for process sequences (Kasperek et al., 2014)

The left side of Figure 4-9 shows the process sequence DSM which forms the basis for the relations within the System Dynamics model shown on the right side. The super-diagonal region of the DSM highlights the feedforward relations among process steps, while its sub-diagonal region shows potential feedback loops in the system. (Browning & Ramasesh, 2007)

The feedforward relationships which represent the precedence relations are modeled in the upper part on the right side of Figure 4-9. The mathematical equation for the starting condition for the particular step can also include conditions such as rates of parallelism between process steps. The lower part represents the feedback loops within the process step sequence which cause rework.

While Figure 4-9 shows the use of composition panels to model relations between elements of the same kind, thus DSM relations, they can also be used to model relations between different kinds of elements, thus DMM relations. For example the allocation of organizational units to process steps can be modeled in the same manner. The framework suggests to use composition panels for relations that are planned to be varied within the System Dynamics simulation environment to observe the effects of the relations and variation as well as to decrease the overall complicatedness of the model.

4.3.2 Step 2b: Quantitative System Dynamics modeling

The System Dynamics modeling classification of engineering design processes and the adaption scheme guide modelers concerning the decision on which stocks, flows, variables and causal loops should be included. After adding these elements to the rework cycle, the structure of the model is established. Additionally, the System Dynamics composition panel supports to structure the relations within the System Dynamics model.

However, the rates or relationships are not quantified by any values or equations yet. Even though, proposals for quantification can be found in the previously listed references the framework for structure-based System Dynamics Analysis comprises three methods and tools to further support the step of quantitative System Dynamics modeling. These methods and tools are heuristics for quantification which can be used as modeling guidelines, specific concepts for handling parameter uncertainty and quantification sheets to increase the reproducibility and accessibility of the System Dynamics models.

Heuristics for quantification

A variety of existing modeling heuristics can be found in literature. DSMs (Design Structure Matrices) are often applied in order to capture phase dependencies (Browning & Eppinger, 2002; Browning, 1998, 2001; Cho & Eppinger, 2005; Eppinger et al., 1994; Yassine et al., 2001) and thus to establish the quantification of process concurrence relationships in System Dynamics models (Ford & Serman, 2003; Kasperek et al., 2014; Le, 2013; Lin et al., 2008; Ruutu et al., 2011). Different quantification approaches for staff allocation are explored by Black & Repenning (2001); Joglekar & Ford (2005); Kasperek, Lindinger et al. (2014); Repenning (2000) and Taylor & Ford (2006)

For the quantification of the influences on productivity and thus, work rates, non-System Dynamics literature may prove helpful (Brunies & Emir, 2001; Kernan et al., 1994; Kvålseth, 1978; Maynard & Hakel, 1997; Nepal et al., 2006; Rosenbaum & Rosenbaum, 1971; Thomas & Napolitan, 1995; Thomas & Raynar, 1997). However, these heuristics from the field of social science have not been applied in System Dynamics models. The studies may still serve as suggestions for the quantification. This shows that the integration of other disciplines outside

of System Dynamics can be useful for the creation of sophisticated models. Bradley et al. (2013) have already discovered that System Dynamics incorporates exchange with other disciplines.

The question remains how researchers gain the information for quantifying the relationships and factors. In most articles that provide information on data acquisition, the answer is interviews with employees (Ford & Sterman, 1998a; Le, 2013; Oliva, 2003). Le et al. (2012), Lin (2006) and Ruutu et al. (2011) provide information concerning the length as well as the number and type of participating employees for the interviews or workshops.

As a rough rule of thumb, heuristics on the general “how to” of quantification can be found in System Dynamics literature, while information in specific quantification approaches and values is more likely to be found in the specific research domains such as engineering design. Other modeling disciplines can rather provide out of the box approaches or different views on quantification such as Brook’s law on software engineering (F. P. Brooks, 1975): “Adding manpower to a late software project makes it later.”

Facing the amount of heuristics on various level of abstractions that have been proposed within various research domains and the fact that the heuristics of interest for modelers are different for each modeling activity, in the following three sets of heuristics are presented that prove especially helpful while developing the structure-based System Dynamics Analysis framework:

- general heuristics for modeling complex systems (Biemans et al., 2001)
- heuristics for engineering design processes (Suss & Thomson, 2012)
- challenges for System Dynamics modeling (Sterman, 2000).

General heuristics for modeling complex systems

Biemans et al. (2001) provide a set of heuristics for dealing with the complexity of business system architecting which can also be applied to quantitative System Dynamics models. Table 4-10 shows the heuristics.

Table 4-10 *General heuristics for modeling complex systems (Biemans et al., 2001)*

Topic	No.	Heuristic
Pursuing high-quality models	G1	Conceptual integrity of models eases their understanding.
	G2	Document (in the model) the reason or source of irrational decisions, as well as the way they can be "solved" when circumstances change.
	G3	First go for an unambiguous model, only then try to be (more) correct or complete.
Syntax and ease of understanding	G4	Use domain-specific terminology, notation, and conventions.
Structuring and abstraction levels	G5	Use a limited number of predefined abstraction levels.
	G6	The choice of abstraction levels should be an opportunistic, domain-specific one.
	G7	The expressiveness of the modeling language should be enforced by the business design objectives, not the other way round.
	G8	The relationship between abstraction levels should be explicitly modeled for each model element, using notation with a formal semantics.
	G9	Restrict a model editing session to a single "type" of refinement.
The process of modeling	G10	For a proper understanding of a model element, model its environment (context) as well.
	G11	The architect should know the usage context of a model to avoid modeling irrelevant details.
Accumulating experience	G12	Besides reusing existing solutions, the value of patterns or reference models is that they give a clue for the appropriate level of abstraction.

Heuristics for engineering design processes

Even though the set of heuristics of Biemans et al. (2001) can be applied to engineering design processes, due to their specific characteristics, special heuristics for engineering design processes can be stated. Based on a literature review Suss & Thomson (2012) provide numerous heuristics for engineering design processes which are described in the following Table 4-11 to Table 4-19.

They structure the heuristics found in the literature review in eight categories and provide additional new heuristics. Table 4-11 presents the first set of heuristics which consider the organization theory and the information processing in the context of engineering design processes.

Table 4-11 *Heuristics considering organization theory and the information processing (Suss & Thomson, 2012)*

No.	Heuristics considering organization theory and the information processing
H1	As the amount of uncertainty increases, the frequency of non-routine, unique, consequential events which cannot be foreseen and which require decisions to be made increases.
H2	Making decisions requires information gathering and communication about the state of affairs that led to the events. This is referred to as information processing. Information processing takes time and requires resources.
H3	Increasing uncertainty increases workloads and time delays in the decision-making mechanisms. Better coordination of information processing mitigates increasing workloads and time delays.
H4	According coordination theory, dependencies among activities and resources create problems that constrain how activities can be performed.
H5	Differentiate between two different kinds of coordination: The tactical coordination during the execution of a process to ensure that each subtask reliably has what it needs, when it needs it. The strategic coordination, where the structure of a process, the information processing, and decision-making mechanisms are designed in a way that the execution of each subtask can take place in concert with the other subtasks with minimum effort and minimum delay.
H6	Use the strategic guidelines of (Pahl & Beitz, 1996) that split the design process into four main phases. Within a phase, much of the work of design consists of proposing solutions for specific design requirements and to evaluate whether or not the solutions meet the criteria developed for the design problem. Iteration between phases is described in terms of the evaluation and feedback that take place within each stage.
H7	During the design review at the end of each phase, there is an opportunity not only to evaluate solution specifications in light of product requirements, but also to synchronize the remaining work for all those participating in the design project.

Table 4-12 presents the second set of heuristics which consider iteration in the context of engineering design processes. Next to the statement that engineering design processes are often iterative (H8), the second heuristic H9 further details what iteration is and where it comes from. The third heuristic H10 tries to draw a picture on how to understand iterative behavior.

Table 4-12 *Heuristics considering iteration in engineering design (Suss & Thomson, 2012)*

No.	Heuristics considering iteration in engineering design
H8	Complex engineering design processes are often iterative.
H9	Design iteration implies rework or refinement of activities to account for changes in the activities' inputs. The changes can be caused by additional or changed information or failure to meet design requirements. (Browning, 1998; R. P. Smith & Eppinger, 1997)
H10	Iteration in engineering design can be seen as a cyclical process that achieves a succession of intermediate improvements on the way towards a final outcome rather than a strictly incremental process that arrives at a single final outcome. (Safoutin, 2003)

Table 4-13 presents the third set of heuristics which considers the effect of uncertainty in engineering design. According to H11 it should be distinguished between epistemic and aleatory uncertainty. This implies that there is a significant amount of aleatory, thus, random uncertainty in the context of engineering design processes which has to be accepted and dealt

with. H12 states that variability which is a sub group of epistemic uncertainty can be reduced by frequent information exchange between related elements.

Table 4-13 Heuristics considering the effect of uncertainty in engineering design (Suss & Thomson, 2012)

No.	Heuristics considering the effect of uncertainty in engineering design
H11	Distinguish between the lack of knowledge (epistemic) and random (aleatory) uncertainty (Oberkamp et al., 2004).
H12	The influence of variability can be managed through frequent information exchange between dependent activities.

Table 4-14 presents the fourth set of heuristics which considers the definition of the state of the task progress in context of engineering design processes. The heuristics connect the state of task progress with the level of present uncertainty in the sense that uncertainty decreases with increasing task progress. The heuristics H16 and H17 suggest the S-shape function as modeling construct for the decrease of uncertainty, thus increase of task progress.

Table 4-14 Heuristics considering defining the state of task progress (Suss & Thomson, 2012)

No.	Heuristics considering defining the state of task progress
H13	Work progress can be seen as a process in which the level of uncertainty in the artefact is reduced as the design progresses. (Hykin & Laming, 1975)
H14	Engineering design can be seen as a process in which the level of design imprecision is reduced, although a degree of stochastic uncertainty usually remains. (Wood et al., 1990)
H15	If engineering design is seen as a process of generating information to reduce uncertainty, the rate of progress of the information developed by a task in engineering design is measured by how it reduces epistemic uncertainty.
H16	Learning or discovery of information required to complete the task reduces the epistemic uncertainty in a task. If so it exhibits a slower rate of change at the beginning and end of the task as in a sigmoid or S-shaped function (Ritter & Schooler, 2002).
H17	The S-shape is the best shape function to represent task completion as it can embody a growth phenomenon similar to learning. (Carrascosa et al., 1998)

Table 4-15 presents the fifth set of heuristics which considers quality in engineering design. H18 and H19 represent two different views on quality and are to a certain extent contradictory. On a more operational level H20 states what increases quality.

Table 4-15 Heuristics considering quality in engineering design (Suss & Thomson, 2012)

No.	Heuristics considering quality in engineering design
H18	In the conventional understanding, higher quality in engineering design can be achieved only at the expense of increased development expenditures and longer cycle times. (Harter et al., 2000)
H19	Alternatively quality, cost, and cycle time in engineering design can be seen as complementary, that is, improvements in quality directly relate to improved cycle time and productivity. (Nandakumar et al., 1993)
H20	Improvements in quality arise from reduced defects and rework in a mature engineering design process (Harter et al., 2000) and better knowledge sharing, acquisition, integration, and application in new PD processes (Jing & Yang, 2009).

Table 4-16 presents the sixth set of heuristics which considers the complexity of engineering design processes. All three heuristics suggest that the modeling of process complexity should be based on or confirmed by empirical observations.

Table 4-16 Heuristics considering complexity of engineering design processes (Suss & Thomson, 2012)

No.	Heuristics considering complexity of engineering design processes
H21	The modeled processes can be based on empirical observation through either the functional forms or the parameter settings or both.
H22	If empirical estimates are not available, empirical work may still provide much information for model construction, and variations and sensitivity analysis can be used to examine the robustness of the results.
H23	The results of the simulation can either be empirically grounded through comparison with observations of real systems or serve as a basis for subsequent observations of these systems.

Table 4-17 presents the seventh set of heuristics which considers communication and system-level integration in the context of engineering design processes. The first two heuristics focus on communication in collaborative engineering environments for which system-level integration is typical. H26, H27 and H28 focus on difficulties of communication with the increasing number of communication channels.

Table 4-17 Heuristics considering communication and system-level integration (Suss & Thomson, 2012)

No.	Heuristics considering communication and system-level integration
H24	Collaborative engineering design is a series of multi-phase design processes linked via the original equipment manufacturer, which acts in a project as the system-level integrator of the various subsystems. The system-level integrator transmits requirements to information as well as he receives information directly and transmits requests for information from local-level development teams. (Yassine & Braha, 2003)
H25	Direct communication between local-level development teams is often informal and unscheduled. The rate at which this information is received, processed, and transmitted between the system-level and the local-level teams is an important element in the rate of progress of the overall design process (Mihm et al., 2003; Yassine et al., 2003).
H26	Communication channels tend to become overloaded and thereby cause delays in the progress of the project. Actor who lack the requisite information usually obtained through hierarchical channels tend to take default decisions.
H27	The following reasons are typical difficulties in large-scale engineering design information management: diversity of channels, scale, variety of perspectives, and uncertainty (Eckert et al., 2001).
H28	The interaction of communication with planning and product complexity has been observed to lead to wasted effort and rework in the case of automotive engineering design. (Maier et al., 2008)

Table 4-18 presents the eighth set of heuristics which considers models of engineering design processes. H29 connects the reduction of uncertainty to the evolution of information. In contrast to previous heuristics, here the degree of uncertainty is not seen as a measure of the task

progress, but as driving factor the task progress. H30 and H31 provide guidance on how to model and quantify rework in the context of engineering design processes.

Table 4-18 Heuristics considering models of engineering design processes (Suss & Thomson, 2012)

No.	Heuristics considering models of engineering design
H29	The evolution of information is roughly equivalent to the rate of reduction of uncertainty. If uncertainty is reduced quickly, the information generated by the task evolves quickly towards its final value. (Krishnan, 1996)
H30	Relationships of functional interaction and overlapping of activities on effort and total span time using empirically derived relationships between rework and the level of functional interaction can be found in (Bhuiyan et al., 2004).
H31	For most models focusing on the mechanisms of iteration and rework that occur in engineering design, the probability of rework was determined a priori as an input to the model. They generally assume a priori levels of rework based on empirical data gathered in organizations with fixed coordination schemes employed.

Based on their own findings, Suss & Thomson (2012) propose the heuristics presented in Table 4-19 to reduce the project span time and effort of engineering design processes. In contrast to the previous heuristics, these ones can be more interpreted as standard operation procedures.

Table 4-19 Heuristics to reduce project span time and effort of engineering design processes (Suss & Thomson, 2012)

No.	Heuristics considering to reduce project span time and effort of engineering design processes
H32	Overlap sequentially dependent tasks when there is sufficiently frequent, interim information exchange to the extent warranted by rates of reduction of epistemic and aleatory uncertainty.
H33	Structure projects such that the size of interdependent tasks is similar. All tasks wait for information generated by tasks that take a longer time and thus are held to the rate of progress of the slowest task.
H34	Provide sufficient resource capacity early enough by anticipating workload requirements. Insufficient resource capacity in critical tasks impedes information flow and causes other tasks to be starved or to receive information out of synchronization and thus perform more rework.
H35	Implement 'scrum' methods where there is high uncertainty to enable intense coordination and keep PD projects on track. This reduces latency delays of information exchange most effectively, forces synchronization of information more frequently, and allows project leaders to review the work done in each sprint, to maintain the influence of the overall organization, and to provide guidance.
H36	Adopt policies and implement systems to reduce delays and to reduce effort in communication of information in the information flow between dependent tasks

Challenges for System Dynamics modeling

Following the general heuristics on the modeling of complex systems and the heuristics on engineering design processes, additional heuristics can be found within the System Dynamics research community. An extensive collection of information on System Dynamics can be found in Sterman (2000). It comprises a large set of heuristics and models which can be taken as a basis for quantification. In particular Sterman explains 79 challenges of System Dynamics and

provides potential solutions in form of heuristics and examples. In the following the three challenges which may be most relevant are described.

Identifying feedback structure from system behavior: Sterman (2000) suggests to first identify and sketch all positive loops which are responsible for growth and all negative loops which halt growth. Then all negative loops that might be responsible for the goal-seeking (converging to a terminal value) behavior should be identified as well as the state of the system, the goal, and the corrective action(s) for each case. As third step negative loops and time delays that might be responsible for oscillations should be identified and again the state of the system, the goal, the corrective action, and delays.

Linking stock and flow structure with feedback: Using the example of the US gasoline shortages in the 1970s, Sterman (2000) suggests that it may be necessary to model the whole system. Additionally information inputs to the rates of flow in the model should be identified and used to close feedback loops. These loops are crucial for the system behavior. Furthermore learning effects should also be considered and modeled if necessary.

Process Improvement: Sterman (2000) suggests to conduct interviews and use this information to develop a single causal diagram capturing the dynamics of the system. The loops within the diagram should be named with the terms used by the interviewees where possible. Additionally each loop should be explained in a paragraph or two capture the dynamics and the link to the interviews. Where possible, the model should reflect the basic physical structure of the system.

Appendix 9.2 lists 66 of these challenges which may be most relevant for the approach presented in this thesis. As the solutions are described in detail in Sterman (2000), this thesis refers to original reference for detailed information on the other particular heuristics.

Modeling concepts for parameter uncertainty

Next it is important to understand the relationship between structure and behavior of complex dynamic systems. Sensitivity analysis helps to test the robustness of the simulation results with respect to uncertainty in the estimated parameters. Additionally, it guides the necessary data collection efforts. As parameters can never be estimated perfectly, it must be decided which to focus on and when to stop. If a sensitivity analysis reveals that a parameter strongly affects the simulated behavior, additional data may be collected leading to a better and more reliable simulation. On the other hand, parameters that show little effects in the sensitivity analysis should not be further investigated as the simulation results are robust even with an approximate estimate. Furthermore, parameters that strongly affect the simulation results may be tuning parameters for process optimization. (Sterman, 2000)

Sterman (2000) suggests to write down the expectations before conducting the particular sensitivity analyses. After each analysis the simulated outcome should be compared to the expected outcome. The explanations should be reformulated for the effect of each change and compared to the simulation results again. As a rule of thumb for the variation of parameters, it should be varied a bit more as the expected likely range of uncertainty: Interviewees tend to be overconfident and underestimate effects. Additionally, the test of extreme values might be insightful. Even though it is important to vary each parameter in isolation to understand parameter sensitivity, it should be kept in mind that engineering design processes are complex

non-linear systems. This means that results of such a univariate sensitivity analysis may provide limited guidance to the response of the engineering design process to multiple parameter changes. To understand the full range of responses of the system with respect to parameter uncertainty a multivariate sensitivity analysis may have to be conducted. (Sterman, 2000)

An additional approach to cope with parameter uncertainty are multi-run simulations. Thereby, the simulation is run with different parameter values multiple times. This allows to replace uncertain parameters by distribution functions such as the Gaussian distribution. As common System Dynamics software packages possess interfaces to data processing systems, multi-run simulations can be automated and the simulation results for the particular runs can be exported. The outcomes of the different simulation runs can be illustrated in one figure in relation to the varied parameter. Figure 4-10 shows the exemplary illustration of multi-run simulation results taken from Petz et al. (2014). Illustrations as such may allow to increase the understanding of dependencies between different parameter values.

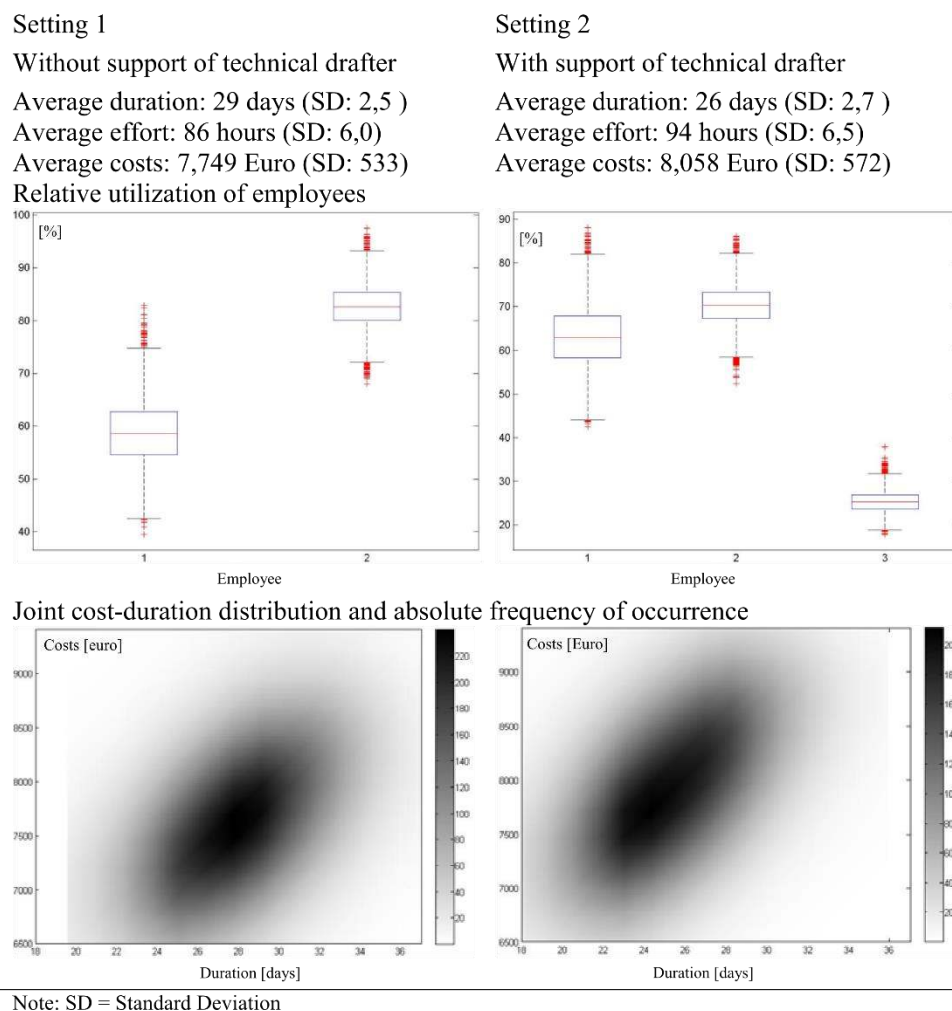


Figure 4-10

Exemplary illustration of multi-run simulation results taken from Petz et al. (2014, p. 164)

Quantification sheets

The structure-based System Dynamics approach proposes the development of two models: The MDM and the System Dynamics model. While the data stored in the MDM is usually easily accessible, especially the data and assumptions used for the quantification are well hidden within the System Dynamics model. Within the case studies this information was identified as very important and should be reproducibly documented. Even though most of modellers have an aversion against documentation (see Kasperek et al. (2015) for a review), documentation is important as otherwise it is difficult to decide for others than the modelers themselves if the model at hand is sufficient in quality, scope and underlying information for the desired purpose (Kasperek, Maisenbacher, Kohn, et al., 2015).

To document this information the creation of quantification sheets is suggested. As for example Clevenger et al. (2012) state that documentation effort should be not excessive, quantification sheets can be simple tables which store the following information for each quantified variable and parameter:

- variable
- equation or value
- unit
- explanations and assumptions
- source of information.

Figure 4-11 shows an exemplary excerpt of a quantification sheet.

Variable	Equation	Unit	Explanation /assumptions	Source
Original Work to do	= - Progress rate	Work package (WP)	The original amount of work is decreased by the work done for each time step (=progress rate)	Common sense
Progress rate	= IF THEN ELSE ("Start Task" = 1 :AND: "Original Work to do" >= 0 :AND: "Allocated Org. Units" > 0 :AND: "Used Tools" > 0 :AND: "Used Methods" > 0, 1.5, 0)	WP / t	If the necessary starting condition "Start Task" is true and there is still work to do and organizational units as well as tools and methods are allocated to the process step, the progress rate is set to 1.5. Otherwise it is set to 0.	Major modeling assumption based on discussions with the project management
...

Figure 4-11 Exemplary quantification sheet

4.4 Step 3: Testing of the System Dynamics model

The general purpose of testing is verifying the overall suitability of models for their purpose, their conformance to fundamental formulation principles, the sensitivity of results to uncertainty in assumptions and the integrity of the modeling processes. Over time, System Dynamics modelers have developed a wide variety of specific tests to uncover flaws and improve models. (Sterman, 2000)

In the context of model testing it has to be distinguished between model calibration, verification and validation. Model calibration should be seen as part of the previous step of quantitative

System Dynamic modeling. The step of testing the System Dynamics model focuses on establishing a degree of model credibility in the sense of confidence in the model. Therefore, depending on the particular conditions, verification, validation tests or both should be conducted.

During verification an apparently correct model is exercised for the specific purpose of finding and fixing modeling errors. It refers to the processes and techniques that are used to assure that the model is correct and matches any agreed-upon specifications and assumptions. During validation the modelers and people knowledgeable of the real system or new/modified design jointly work to review and evaluate how the model works. It refers to the processes and techniques jointly used by the modelers, model customer and decision makers to assure that the model represents the real system (or proposed real system) to a sufficient level of accuracy. (Carson, 2002)

It needs to be noted that no model is ever 100% verified or validated. As models are representations of systems and the behavior of the models is at best an approximation to the system's behavior. When a model is referred to as verified or validated, it is meant that a series of tasks to verify and validate was explicitly carried out to the degree necessary for our purposes.

Sterman's verification tests

As tool and method for the testing step of the structure-based System Dynamics Analysis framework the collection of verification criteria introduced by Sterman (2000) is suggested. Table 4-20 to Table 4-22 summarize the main tests, the purpose of each, as well as the principal tools and methods used in each.

Table 4-20 Tests for verification of dynamic models (Sterman, 2000), part 1

Test	Purpose of Test	Tools and Procedures
1. Boundary Adequacy	<p>Are the important concepts for addressing the problem endogenous to the model?</p> <p>Does the behavior of the model change significantly when boundary assumptions are relaxed?</p> <p>Do the policy recommendations change when the model boundary is extended?</p>	<p>Use model boundary charts, subsystem diagrams, causal diagrams, stock and flow maps, and direct inspection of model equations.</p> <p>Use interviews, workshops to solicit expert opinion, archival materials, review of literature, direct inspection/participation in system processes, etc.</p> <p>Modify model to include plausible additional structure; make constants and exogenous variables endogenous, then repeat sensitivity and policy analysis.</p>
2. Structure Assessment	<p>Is the model structure consistent with relevant descriptive knowledge of the system?</p> <p>Is the level of aggregation appropriate?</p> <p>Does the model conform to basic physical laws such as conservation laws?</p> <p>Do the decision rules capture the behavior of the actors in the system?</p>	<p>Use policy structure diagrams, causal diagrams, stock and flow maps and direct inspection of model equations.</p> <p>Use interviews, workshops to solicit expert opinion, archival materials, direct inspection or participation in system processes, as in (1) above.</p> <p>Conduct partial model tests of the intended rationality of decision rules.</p> <p>Conduct laboratory experiments to elicit mental models and decision rules of system participants.</p> <p>Develop disaggregate submodels and compare behavior to aggregate formulations.</p> <p>Disaggregate suspect structures, then repeat sensitivity and policy analysis.</p>
3. Dimensional Consistency	<p>Is each equation dimensionally consistent without the use of parameters having no real world meaning?</p>	<p>Use dimensional analysis software.</p> <p>Inspect model equations for suspect parameters.</p>
4. Parameter Assessment	<p>Are the parameter values consistent with relevant descriptive and numerical knowledge of the system?</p> <p>Do all parameters have real world counterparts?</p>	<p>Use statistical methods to estimate parameters (wide range of methods available).</p> <p>Use partial model tests to calibrate subsystems. Use judgmental methods based on interviews, expert opinion, focus groups, archival materials, direct experience, etc. (as above)</p> <p>Develop disaggregate submodels to estimate relationships for use in more aggregate models.</p>

Table 4-21 Tests for verification of dynamic models (Sterman, 2000), part 2

Test	Purpose of Test	Tools and Procedures
5. Extreme Conditions	<p>Does each equation make sense even when its inputs take on extreme values?</p> <p>Does the model respond plausibly when subjected to extreme policies, shocks, and parameters?</p>	<p>Inspect each equation.</p> <p>Test response to extreme values of each input, alone and in combination.</p> <p>Subject model to large shocks and extreme conditions.</p> <p>Implement tests that examine conformance to basic physical laws (e.g., no inventory, no shipments; no labor, no production).</p>
6. Integration Error	<p>Are the results sensitive to the choice of time step or numerical integration method?</p>	<p>Cut the time step in half and test for changes in behavior.</p> <p>Use different integration methods and test for changes in behavior.</p>
7. Behavior Reproduction	<p>Does the model reproduce the behavior of interest in the system (qualitatively and quantitatively)?</p> <p>Does it endogenously generate the symptoms of difficulty motivating the study?</p> <p>Does the model generate the various modes of behavior observed in the real system?</p> <p>Do the frequencies and phase relationships among the variables match the data?</p>	<p>Compute statistical measures of correspondence between model and data: descriptive statistics (e.g., R2, MAE); time domain methods (e.g., autocorrelation functions); frequency domain methods (e.g., spectral analysis); many others.</p> <p>Compare model output and data qualitatively, including modes of behavior, shape of variables, asymmetries, relative amplitudes and phasing, unusual events.</p> <p>Examine response of model to test inputs, shocks, and noise.</p>
8. Behavior Anomaly	<p>Do anomalous behaviors result when assumptions of the model are changed or deleted?</p>	<p>Zero out key effects (loop knockout analysis).</p> <p>Replace equilibrium assumptions with disequilibrium structures.</p>
9. Family Member	<p>Can the model generate the behavior observed in other instances of the same system?</p>	<p>Calibrate the model to the widest possible range of related systems.</p>
10. Surprise Behavior	<p>Does the model generate previously unobserved or unrecognized behavior?</p> <p>Does the model successfully anticipate the response of the system to novel conditions?</p>	<p>Keep accurate, complete, and dated records of model simulations. Use model to simulate likely future behavior of system.</p> <p>Resolve all discrepancies between model behavior and your understanding of the real system.</p> <p>Document participant and client mental models prior to the start of the modeling effort.</p>

Table 4-22 Tests for verification of dynamic models (Sterman, 2000), part 3

Test	Purpose of Test	Tools and Procedures
11. Sensitivity Analysis	<p><u>Numerical sensitivity</u>: Do the numerical values change significantly...</p> <p><u>Behavioral sensitivity</u>: Do the modes of behavior generated by the model change significantly...</p> <p><u>Policy sensitivity</u>: Do the policy implications change significantly.when assumptions about parameters, boundary, and aggregation are varied over the plausible range of uncertainty?</p>	<p>Perform univariate and multivariate sensitivity analysis.</p> <p>Use analytic methods (linearization, local and global stability analysis, etc.).</p> <p>Conduct model boundary and aggregation tests listed in (1) and (2) above.</p> <p>Use optimization methods to find the best parameters and policies.</p> <p>Use optimization methods to find parameter combinations that generate implausible results or reverse policy outcomes.</p>
12. System Improvement	<p>Did the modeling process help Improvement change the system for the better?</p>	<p>Design instruments in advance to assess the impact of the modeling process on mental models, behavior, and outcomes.</p> <p>Design controlled experiments with treatment and control groups, random assignment, pre-intervention and post-intervention assessment, etc.</p>

4.5 Step 4: Simulative analysis

Based on the verified System Dynamics model the framework highlights three different types of analysis which can be conducted using the framework for structure-based System Dynamics Analysis. The combined consideration of structural and dynamic complexity as well as uncertainty decreases the perceived complicatedness of the considered engineering design process and thus, increases the understanding of the overall complexity. This increased understanding supports risk management by uncovering knowable unknown unknowns which is described in the subsection *Understanding the complexity: Uncovering risks*.

Future conditions: Scenario analysis shows how the created simulation environment allows taking a “what if?” perspective and simulating the effects of potential future conditions. Based on the simulation insights, scenarios can be defined. This includes the impact analysis of events triggered by external or internal factors, the analysis of different scenarios as well as robustness analysis of engineering design processes considering changing boundary conditions.

Furthermore, the created simulation environment can be used to assess the performance of different structural designs of the engineering design process which is described in *Performance of structural designs: Benchmarking*. This benchmarking of different structures is especially interesting for the early planning phase of engineering design processes, as the benchmarking allows insights into which structural design performs best under the assumed conditions.

Understanding the complexity: Uncovering risks

The first of the three highlighted types of analysis is *Understanding the complexity: Uncovering risks*: The structure-based System Dynamics Analysis framework helps understanding the

intrinsic complexity of an engineering design process during its planning phase. Thereby, it considers both dimensions of structural and dynamic complexity and the behavior under uncertainty. This combined consideration of complexity and the ability for experimentation within the MDM and System Dynamics models can be used to uncover previously unknown risks, also referred to as unforeseeable uncertainties or unknown unknowns.

This framework suggests to use the process for diagnosing unforeseeable uncertainty of Loch et al. (2008) as guideline for uncovering risks of engineering design processes with the structure-based System Dynamics Analysis framework, see Figure 4-12. Both frameworks fit well together, as Loch et al. (2008) describe the process steps to uncover risks on a procedural level, while the models of structure-based System Dynamics Analysis framework offer the necessary tools to successfully conduct Loch et al.'s steps.

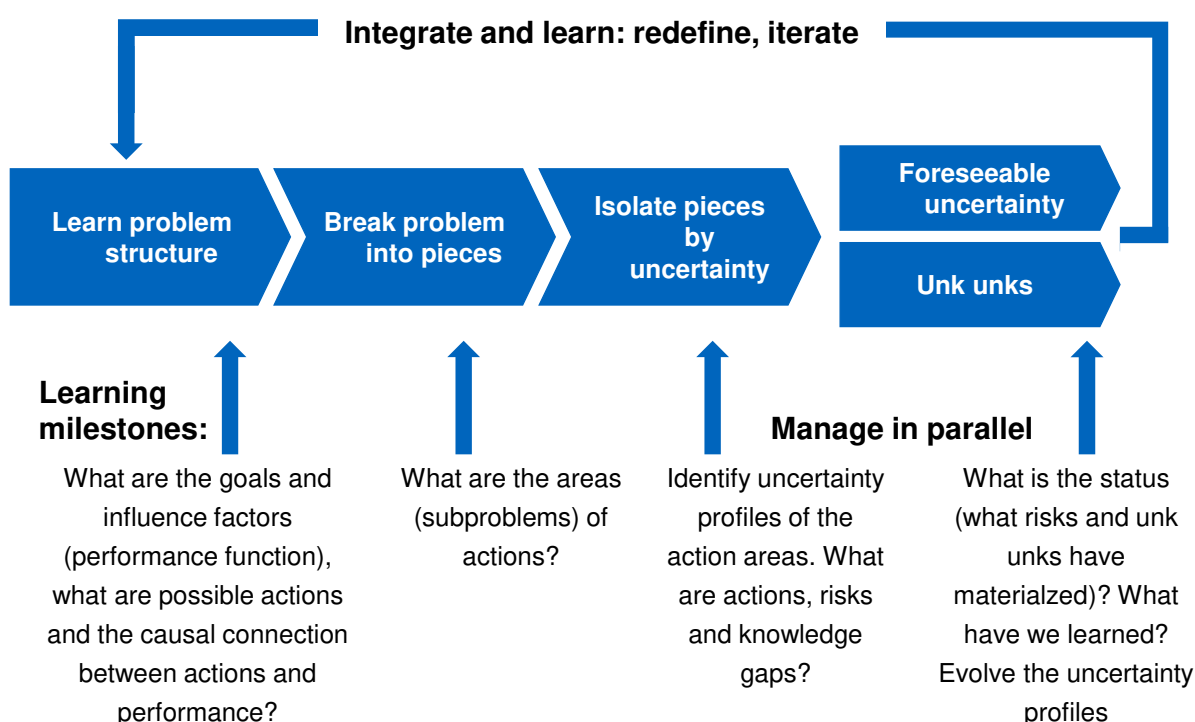


Figure 4-12 A process for diagnosing unforeseeable uncertainty (Loch et al., 2008)

Loch et al. (2008) developed their process for diagnosing unforeseeable uncertainty in a case study with a start-up company. The process comprises the following steps: First, *Learn problem structure* to identify goals, external influences, activities, and causality of activities and effects. Second, *Break problem into pieces* such as process activities, stakeholders, or underlying product modules to delineate the problem. For each problem area typical situations, the types of present uncertainty (foreseeable/ unforeseeable) and their expected severity are documented. Third, *Isolate pieces by uncertainty* to examine the complexity and uncertainty of each piece to identify the major risks that need managing and the knowledge gaps that point to areas of potential unknown risks. This step is a highly iterative and gradual process. Fourth, manage *Foreseeable uncertainty* and *Unknown unknowns (unk unks)* in parallel by using different management approaches according to the uncertainty category: iteration and learning for the

pieces that were threatened by unforeseeable uncertainty, and more structured approaches for the pieces with foreseeable uncertainty.

The process suggested by Loch et al. (2008) can be applied based on the data and the models of the structure-based System Dynamics Analysis framework to uncover risks of engineering design processes. The following paragraphs show how the steps of the process for diagnosing unforeseeable uncertainty can be supported by the steps of the structure-based System Dynamics Analysis.

Process step *Learn problem structure* of Loch et al. (2008) is supported by the step *MDM modeling* of the framework for structure-based System Dynamics Analysis: Both sub-steps of the MDM modeling step, system definition and information acquisition support in learning about the problem structure. Especially the various procedures of the information acquisition described by Lindemann et al. (2009) and adopted for the structure-based System Dynamics Analysis support in how to identify relevant information about the problem. The MDM itself allows to explicitly visualize the relevant domains and interdependencies for the problem such as activities, external influences and causal relations between activities and fosters learning. The MDM metamodel with most common domains and interdependencies serves as a guideline on which aspects to consider within the analysis and explicitly accounts for external influences (events), activities (process steps), causalities (interdependencies and relations).

Process step *Break problem into pieces* of Loch et al. (2008) is supported by the step *MDM modeling* of the framework for structure-based System Dynamics Analysis: One of the strengths of structural system representation in matrix form is its ability of system decomposition (Browning, 2001; Lindemann et al., 2009). In the case of engineering design processes, process activities are illustrated and may be clustered within the MDM, as well as stakeholders may be explicitly modeled. Elements of the domains of the MDM can be chosen for deepened analysis for the type of present uncertainty and its expected severity.

Process step *Isolate pieces by uncertainty* of Loch et al. (2008) is supported by the steps *Transformation and quantification* as well as *System Dynamics analysis* of the framework for structure-based System Dynamics Analysis: The quantification data documented in the quantification sheets can be used as indication of the severity of the present uncertainty. This deepened analysis can vice versa also increase the quality of the quantification values for the simulation. Generally, both, inherent structural and inherent dynamic complexity can be examined with the MDM respectively System Dynamics model derived within the structure-based System Dynamics Analysis. Uncertain relations can be incorporated in the models as well as uncertain parameters can be varied to examine their influence on the engineering design process. Especially the increasing understanding of the interaction of structural and dynamic complexity decreases the perceived complicatedness of the system, thus, may allow to see potential risks which were unforeseeable before.

Process step *Manage foreseeable uncertainty and unk unks in parallel* of Loch et al. (2008) is supported by the steps *System Dynamics analysis* and *Design application* of the framework for structure-based System Dynamics Analysis: Identified foreseeable uncertainty can be examined by scenario analysis which is explained in detail within the next section. However as the available information increases during the course of the engineering design process, the information within the simulation can be regularly updated to analyze the effect on the

simulation outputs. This helps identifying the need for corrective action as soon as possible. Therefore warning and control limits for the simulation outputs can be defined in advance. Additionally to the analysis of the behavioral performance, structural changes to decrease either uncertainty or its effects on the engineering design process can be undertaken as part of the design application step.

Future conditions: Scenario analysis

Scenario planning embodies a set of techniques to deal with uncertainties in an environment by presenting the decision makers several fundamentally different outlooks on the future. It differs fundamentally from forecasting in that it accepts uncertainty, tries to understand it, and makes it a part of the reasoning. Scenarios are not projections, predictions, or preferences; rather they are coherent and credible alternative stories, describing different paths that lead to alternative futures. (Cornelius et al., 2005; Ramasesh & Browning, 2014)

The analysis of multiple scenarios deliberately confronts decision makers with environmental uncertainties by presenting them several different outlooks on the future (Bood & Postma, 1997). Scenarios are generally built upon a sequence of dynamically interacting events, conditions, and changes that are necessary to reach a particular outcome. Consequently, scenarios focus attention on causal processes and crucial decision points. (Cornelius et al., 2005)

Scenarios of future conditions serve multiple functions. According to Cornelius et al. (2005), scenarios:

- Present a background for the design and selection of strategies.
- Help make managers aware of environmental uncertainties by confronting them with fundamentally different future states.
- Provide a tool to identify what might possibly happen and how an organization can act upon or react to future developments. As such, scenarios can be used as early warning systems.
- Offer the possibility to combine quantitative data with qualitative input, enabling scenario planners to incorporate results from other forecasting techniques and allow for soft and fuzzy variables.
- Can help stretch managers' mental models by explicitly confronting them with their own biased viewpoints.

The verified simulation model serves as a platform for experimentation. Different scenarios can be simulated to see how potential future events and conditions of the scenarios affect the engineering design process dynamics. As the System Dynamics model bases on the engineering design process structure, the analyzed future events and conditions can be of structural or dynamic nature. Case study 4 showed an example of scenario analysis with the structure-based System Dynamics Analysis framework. Based on a verified simulation model the effects of varying delivery dates of particular sub systems on the overall construction sequence of a research center were analyzed.

Performance of structural designs: Benchmarking

Proactive design management is one of the key performance indicators for the success of engineering design processes. It is known that the design phase offers the greatest scope for reducing the overall project costs and adding maximum values in the project. Therefore, the size and complexity of modern design with increased uncertainty requires front-end planning of engineering design processes. In recent years, the concept of modeling has become increasingly important in engineering design management. It is not sufficient to pay detailed attention to the design of various elements of a process individually, but all elements must be considered in relation to others in order to make the overall process effective. (Doloi, 2010)

Process simulation may serve as a tool to support engineering design process planning. During the planning phase of engineering design processes various different process sequences may be possible. The structure-based System Dynamics Analysis allows to benchmark the performance of different potential structural process designs by simulation. The benchmarking procedure may support the decision which structural process design to choose under the assumed circumstances and the operational environment.

Figure 4-13 shows a procedure which can be used for simulative benchmarking of different structural designs in the early phase of engineering design processes. The idea of the procedure is loosely based on the “integrated framework for evaluating alternatives and decision analysis process over life of the project” of Doloi (2010).

Doloi (2010) had the idea to analyze the design of projects at an early stage by building hierarchical process models and simulating them by linking the processes and allocating available resources across all disciplines over the lifecycle of projects. He proposed to identify alternative processes and test them for optimal design and overall project configuration.

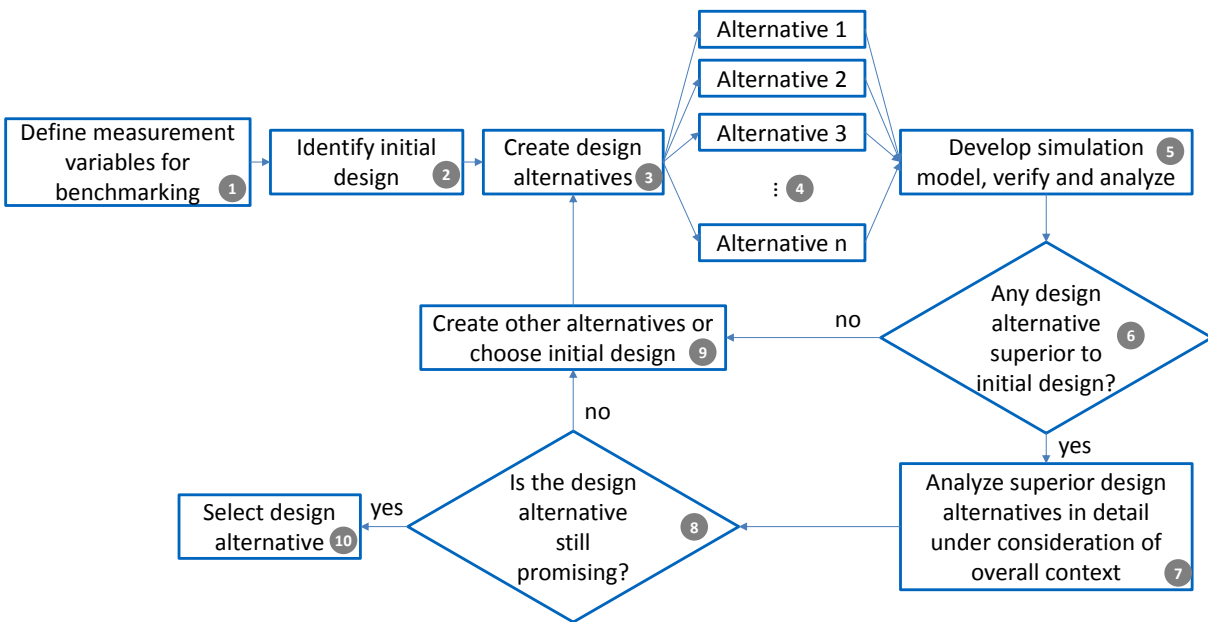


Figure 4-13 Procedure for simulative benchmarking of different structural designs in the early phase of engineering design processes

As first step of the procedure illustrated in Figure 4-13, the measurement variables for the benchmarking of the optimal design have to be defined (1). As basis for the creation of structural design alternatives (3+4) an initial structural process design has to be defined or identified if already available (2). For all design alternatives and the initial design, simulation models have to be set up (5). As the design alternatives usually only differ in particular structural details, large parts of each simulation may be identical. Additionally, a high degree of similarity between the simulation models should be aspired to decrease the potential negative effects of wrong modeling assumptions. If most assumptions are kept constant this potentially negative effect may be canceled down in the relative comparison of design alternatives. After testing that the simulation models sufficiently enough represent reality, the simulation results can be analyzed. If one or more design alternatives show superior results compared to the initial design (6), these design alternatives should be analyzed in more detailed under consideration of the overall process context (7). If after the detailed analysis, a design alternative still seems more promising than the initial design (8), it should be chosen as structural design for the engineering design process (10). If no superior design alternative can be identified by comparing the simulation results, other new alternatives should be created or the initial design should be chosen (9).

4.6 Step 5: Design application

The application of the three analysis directions provides knowledge in understanding the complexity, the system behavior with respect to future conditions and the assessment of different structural designs of the engineering design process from the dynamic complexity dimension through System Dynamics simulation. The construction of the simulation model based on the structural MDM allows a joint consideration of structural and dynamic complexity. Thus, the insights from the simulation can be transferred on the structure of the process.

Complexity analysis report

To ease the application of the insights from the structure-based System Dynamics Analysis on the process structure, the generated information needs to be prepared and sufficiently documented for future use. Within classic structural modeling procedures, no additional information has to be stored by the modelers. Information about why particular domains were chosen, or which sources of information were used is usually not available. However, often this kind of information is necessary to understand the model and its results for anybody else than the modelers itself, because:

- Important decisions made during the modeling process are necessary to understand the model as well as to assess the quality of the model.
- Parts of the model, such as the meaning of domains and dependencies, may not be understandable without further information.
- Information about the background of the modeling activity such as resources and time frame are necessary to assess the model.

Important decisions during the modeling process have to be documented, as well as the reasons why these decisions were made. Independently from the chosen analysis direction, the

generated information with the structure-based System Dynamics Analysis framework is manifold and a procedure for saving this information is needed. This includes the question where and how this information should be documented. In order to enable a practical application of the implications of the structure-based System Dynamics approach on the process structure, this section presents the compilation of a complexity analysis report. The report comprises the:

- underlying complexity challenge
- information of modeling process regarding structural complexity
- information of modeling process regarding dynamic complexity
- MDM and System Dynamics models
- analysis results
- implications on the engineering design process structure and necessary actions.

However, the effort required for applying the complexity analysis report has to be proportional to the benefit of having additional information. For the complete range of possible documentation, from a complete (simplistic) record of all modeling assumptions all the way to no records, the benefit of having additional information changes. The aim of the complexity analysis report is to find a compromise on both scales: effort of documentation and benefit of the information documented. The documentation can require a large amount of additional effort from the modelers and it therefore, can be discussed whether such time-consuming documentation is always required. If the probability of utilization by persons other than the original modelers is low, it might be worthwhile to document less, but for models with a high probability of impact, an entire and structured documentation might be essential. Such documentation can be ensured by using the complexity analysis report.

Underlying complexity challenge

The structure-based System Dynamics approach starts with the system definition as first activity of the MDM modeling step. The result of this step is the so-called metamodel. The metamodel represents the modeler's view of the most important entities and interactions for the description of the considered system (Kortler & Lindemann, 2011). Here, the domains of the MDM and their dependencies are defined.

Without any additional documentation even though an in-depth look into the system, the problem definition and the potential sources of information were conducted, no further information on the considered system than the metamodel itself is documented to this point. This loss of information makes it difficult for others to reproduce the modeling process and the model, as the link between the originating system and the metamodel is very thin. According to Stachowiak (1973) and Kohn (2014), information about the original system is very important to understand the generated model and its functionality.

Therefore, the first part of the complexity analysis report is the statement of the underlying complexity challenge and the model purpose. Both, the complexity challenge and the model purpose can be documented with three to five sentences in continuous text form. However, if the more detailed information is necessary to understand the context, the description can be longer.

Information of modeling process regarding structural complexity

The documented information on the modeling process should include statements about:

- the focus of the model
- sources of information and their assessment
- the chosen technique for information acquisition.

This information focusses on understanding the generation process of the MDM. A statement about the focus of the model is important as all models are simplifications. Therefore, it has to be reproducible which parts of the originating object are considered in the model. This also includes the model boundaries. Additionally, the sources of information have to be explicitly stated and mapped on the areas of consideration to not only understand which information is incorporated in the models, but also where the information came from. As the quality of input usually differs depending on the available types of information and the chosen information acquisition technique, a rough assessment of the information source's quality should be documented as well as the chosen acquisition technique. Common techniques used are: surveys, databases, workshops, etc.

If applicable, the documentation should also include information on why particular areas or sources of information were not considered.

Information of modeling process regarding dynamic complexity

The documented information on the modeling process should include statements about:

- additional focuses of the simulation model
- the chosen abstraction level for the simulation and modeling constructs
- sources of information for quantification and an assessment of them.

This information emphasizes the understanding of the transformation process from MDM to System Dynamics model. Therefore it is necessary to state which additional focuses are considered by the simulation model compared to the MDM. This may include behavioral elements such as the behavior of individuals or the occurrences of particular events. These additional areas of consideration may also influence the qualitative modeling of the System Dynamics model; thus, on which abstraction level the engineering design process is modeled (project, phase or sub phase level) and the particular detailed modeling constructs. Regarding the quantitative modeling of the System Dynamics model, the sources of information for quantification should be stated and also a rough assessment of the information source's quality.

As for the structural modeling process, the documentation should also include information on why particular decisions were made.

MDM and System Dynamics model

The documented information on the models as such should include:

- MDM
- System Dynamics models

- additional descriptions.

Additional descriptions should be documented where information is not self-explaining. This may be descriptions of the:

- domains and interdependencies of the MDM
- elements and relations of the MDM
- stocks, flows and variables of the System Dynamics model
- equations and data of the System Dynamics model.

Especially the understanding of the chosen domains and interdependencies often needs to be supported by additional explanations to understand why objects were modeled the way it was done. If necessary a glossary may support the understanding of elements and relations of the MDM, respectively stocks, flows and variables of the System Dynamics models. The quantification sheets proposed within the framework can be used as form of documentation of equations and data of the System Dynamics model.

Analysis results

The analysis results comprise the simulations of the System Dynamics model. If structural analyses were conducted solely based on the MDM model, these should be also documented here. Each documented analysis should comprise a short description, the results itself and an interpretation of these. The description should comprise the purpose of the particular analysis, the varied parameters and the parameters of interest. If expectations were written down as explained for the handling of parameter uncertainty, these expectations and their deviations from the actual simulation results should also be documented.

Implications on the engineering design process structure and necessary actions

Even though directly based on the analysis results, the implications of the analyses on the engineering design process structure and necessary actions should be documented in a separate section. Especially from a management perspective, only the outcome of the analysis may be of interest. A separate section eases the finding of these and therefore may increase its impact. As the implications and necessary actions may be read individually from the rest of the complexity analysis report, they should be understandable without the context of the overall report. However, the implications should be traceable to the conducted analyses suggesting them.

5. Evaluation of the framework

The previous chapters elaborated on characteristics of and modeling approaches for engineering design processes as well as the case studies which led to the development of the framework for structure-based System Dynamics Analysis of Engineering Design Processes. Consequently the framework with its steps, supporting tools and methods was introduced and summarized. To complete the approach, this chapter provides a validation example of the framework as descriptive study II following the previous prescriptive study in chapter 4. The presented case study alone cannot cover all aspects of the framework, yet it can point out the focal ideas and its general practicability and validity. It is to be considered as one of many case studies conducted during and after definition of the overall framework for the structure-based System Dynamics Analysis of Engineering Design Processes.

5.1 Initial situation and description of the use case

As an introduction to the evaluation example, the following sections roughly outline the situation at the industry partner, the use case itself and the objectives as well as instruments of the use case.

The use case was conducted in form of a student project (Produktentwicklung, 2015) and bases on a submitted publication (Kasperek et al., 2016).

A subset of the engineering design process at a large German manufacturer of commercial vehicles was selected as a use case. The company produces a wide portfolio of both trucks and buses out of a rule-based modular kit that enables the reuse of components across the product portfolio to minimize the technical complexity (Kreimeyer et al., 2014).

The subset for the use case plays a central role in the engineering design process and provides a good example for the dynamics at the relevant company. It is the change management process of the variant options and rules that govern the technical description of the modular kit in the company's data management systems (especially PLM, bill of material, and CAx data management). The decision to choose this subset was verified with the senior management of the company. The subset is important, as only a consistent set of variant options allows for configuring different products out of a generic pool of modules (Tidstam & Malmqvist, 2015). As the variant options change during the process and become more mature before being released (e.g. from "fuel range min. 300 km per day" to "460 l fuel tank"), the rules based on this description need to be constantly adapted, and the associated bill of material evolves in parallel.

The change management process of the variant options and rules is led by product architects and conducted together with sales experts, product management experts and master data managers.

The use case is described following the steps of the framework for structure-based System Dynamics Analysis of Engineering Design Processes. However, also the five stages for conducting case studies of Stuart et al. (2002) could have been used to guide through the use case. While their stages focus on the execution of the case study, the framework for structure-

based System Dynamics Analysis of Engineering Design Processes focuses on the modeling perspective. As the use case reflects the evaluation of the framework, its description is based on the steps of the framework.

5.2 Objectives and instrumental development

Objectives

The industry partner was interested in optimizing the early phase of the engineering design process. By "early phase" the industry partner refers to the phase from the beginning of the engineering design process to the decision to develop the concept into a product, which aligns with the definition of early phase within this thesis. The interest particularly consists of finding out how the process can be adapted to improve the resource allocation and the operations of the company's internal committees that govern the exchange of information throughout the distributed design process.

On a more detailed level, the industry partner had the following questions:

- What factors influence the process and what effects do they have?
- Where does significant resource consumption occur and what is its origin?
- Is it possible to reduce the time of the given process, and if so, which influencing factors have to be adapted in what way?
- How robust is the process against changes? In other words, which parts of the process are most susceptible to possible external disturbances (e.g. through bad data quality or delays in the inputs delivered)?
- How can different improvement scenarios be assessed and compared to identify the optimal one?

The questions and targets for the analysis – for the relevant process – were collected and reviewed with the team responsible for the process. Furthermore, the questions were confirmed with the management as a valid goal for an in-depth analysis and review. This allowed to address the basic question from the research perspective:

Can the framework for structure-based System Dynamics Analysis be applied in an industrial context and does it successfully serve as a tool to analyze engineering design processes and deduce improvement actions?

Referring to Design Research Methodology of Blessing & Chakrabarti (2009), the first part of the research question can be seen as application evaluation, while the second part can be seen as initial success evaluation. On a more detailed level, the second part of the question refers to the desired contributions of the framework as stated in introductory chapter; in particular, if the framework gives decision support for the design of engineering design processes through simulation experiments and through building scenarios.

Instrumental development

During the six month period of the application one researcher worked at the site of the industry partner as part of the architecture group whose members operate the considered process. As the simulation was iteratively developed over time, selected process experts of the industry partner were regularly informed about the current status of the structural and System Dynamics

simulation and gave feedback. This multi-faceted approach to gain the complete picture is consistent with the ethnographic data gathering approach and concepts of triangulation (Jick, 1979; Schein, 1990; Stuart et al., 2002).

Table 5-1 gives an overview of the steps that had to be conducted in the use case. For each step specific persons in charge were assigned. Except for the first three rows, the detailed steps are assigned the five steps of the framework. The first three rows are be assigned to steps of the framework as they do not focus on modeling but on the definition of objectives for the use case. Additionally, the Table 5-1 indicates the person who reviewed each step's outcome at the company.

Table 5-1 Overview of the specific steps to be conducted within the case study

Overview of research evaluation and validation	Work completed by	Work reviewed in company by (role / number of persons)
Definition of goals and scope of the analysis	researcher	architecture expert
Design of basic concept and scope of process model and analysis	researcher, architecture expert	two product architecture experts
Review of goals and scope of analysis	researcher	two product architecture experts, senior management product architecture
MDM modeling: setup of MDM	researcher, architecture expert	two product architecture experts
MDM modeling: validation of MDM	researcher	five product architecture and bill of material experts, senior management product architecture and sales master data
Transformation & quantification: Quantitative SD modeling	researcher	two product architecture experts
Transformation & quantification: Qualitative SD modeling	researcher, architecture expert	two product architecture experts
Testing of the System Dynamics model	researcher	five product architecture and bill of material experts, senior management product architecture and sales master data
Simulative analysis: Future conditions - generation of scenarios	researcher, architecture expert	two product architecture experts
Simulative analysis: Future conditions - validation of scenarios	researcher	two product architecture experts, senior management product architecture, senior management bill of material
Design application: Review of complexity analysis	researcher	product architecture team, senior management (four managers), vice president engineering

5.3 Step1: MDM modeling

The domains of the MDM were chosen according to the MDM metamodel of the framework. With respect to the purpose of the use case and the desired results, the dependencies between the domains were defined as illustrated in Figure 5-1. The model served as an overview of the process and the stakeholders involved. It was built based on the company's process description, which turned out to be rather vague. It had to be refined based on interviews with the persons

As indicated in Table 5-1, both the metamodel and the detailed MDM were verified with five product architecture and bill of material experts, senior management product architecture and sales master data.

5.4 Step 2a: Qualitative System Dynamics modeling

Figure 5-3 illustrates the qualitative System Dynamics model. The stocks and flows of the model were derived from the process step DSM (“process step is followed by process step”) within the MDM of Figure 5-2. As the process considered is only a small part of the overall engineering design process of the industry partner, the description of its detailed process steps was considered to be sub-phase level. Consequently, stock+flow constructs were chosen to represent the particular process steps. The flows therefore represent the relations between the process steps given in the MDM. In addition, the model contains a source for incoming and a sink for declined requests.

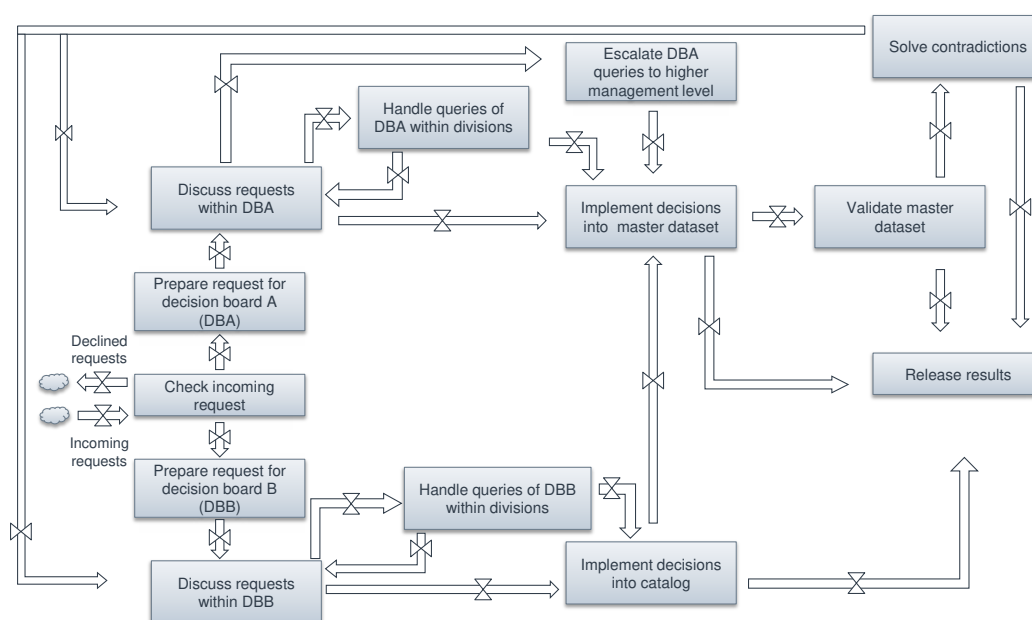


Figure 5-3 Qualitative System Dynamics model of the process

The qualitative System Dynamics model illustrates the sequence of process steps, while the MDM also represents all other qualitative data, such as organizational units, resources, events and time estimates.

5.5 Step 2b: Quantitative System Dynamics modeling

First, the quantitative variables to be included in the model were identified. They depend on the objectives of the simulation, which result from the research questions. Table 5-2 shows the mapping of the research questions derived from the objectives of the use case and the established quantitative variables for answering them.

Table 5-2 Quantitative Variables for answering the research questions

Research questions	Quantitative variables
1. Which factors influence the process and what specific consequences do they have?	<ul style="list-style-type: none"> • Influencing factors
2. At which point in the process does a high consumption of resources occur and how can it be avoided?	<ul style="list-style-type: none"> • Time effort of the department of product architecture
3. By which means can the cycle time of the given development process be reduced and at which scale do these means operate?	<ul style="list-style-type: none"> • Time effort of the department of product architecture • Waiting time of process elements
4. What effects does the change of constraints have on the process?	<ul style="list-style-type: none"> • Temporal dependency of the arriving queries (query-input-function)

To answer the question of which factors influence the process, influencing factors were introduced. A total of nine significant influencing factors were identified and are listed below. There, the terms DBA and DBB refer to the decision boards A and B. The influencing factors were obtained through expert interviews, participation in the process and review of additional documents (protocols, agendas) from process operations in the past:

- frequency decision board A (DBA)
- frequency decision board B (DBB)
- frequency post-processing
- Quality Gate
- probability of iteration (DBA)
- probability of escalation (DBA)
- probability of iteration (DBB)
- maximum workload (DBA)
- maximum workload (DBB).

The time effort of each process step was modeled in the System Dynamics model to analyze resource consumption. When branches existed in the model, the corresponding flow probabilities were identified through interviews.

Figure 5-4 shows the simulation model in the software tool Vensim®. The process steps are represented by stocks and their dependencies by flows. The green boxes represent the influencing factors described above. The time effort by the product architecture group and the waiting times are directly implemented in the stocks and flows. The blue arrows indicate dependencies between the elements.

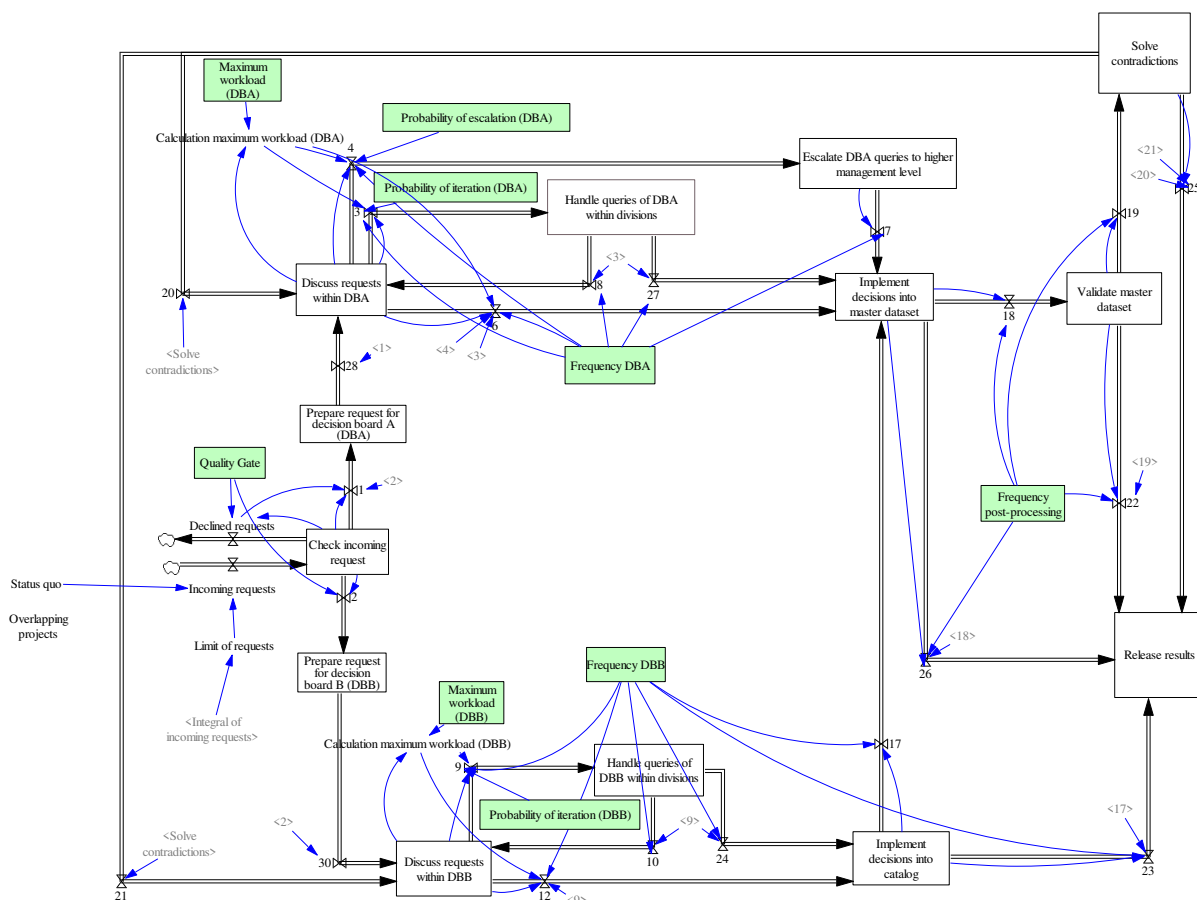


Figure 5-4 System Dynamics model of the industrial engineering design process in Vensim® software

Figure 5-5 shows an exemplary simulation result of Run_Nr_763 in the software. The amount of incoming requests to be checked per day is illustrated in blue, and the total amount of released results over time in red.

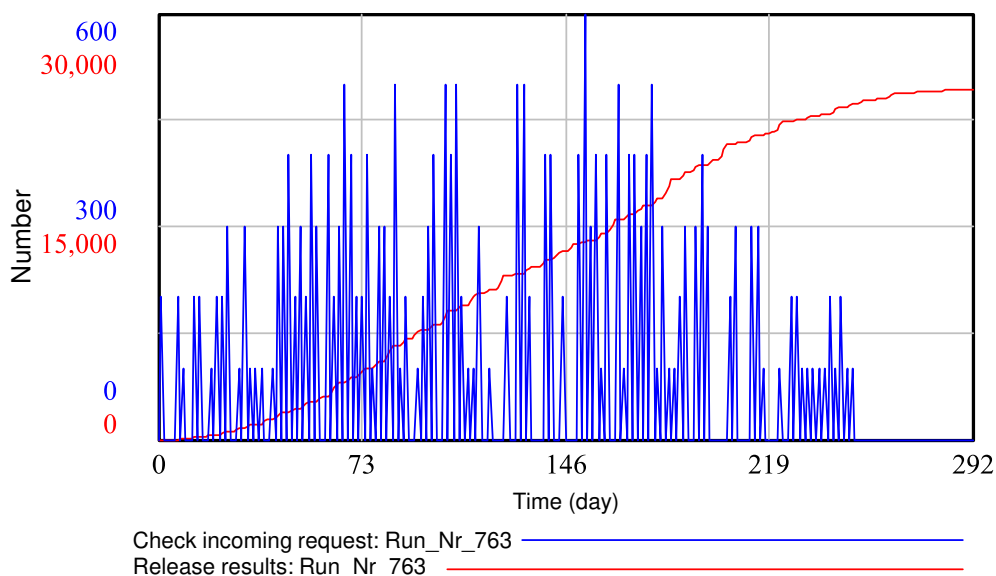


Figure 5-5 Exemplary simulation results of incoming requests and released results in Vensim®

5.6 Step 3: Testing of the model

Based on the simulation, the applied parameters were adapted to validate the model and ensure that the experts involved recognized the actual process in the model. This was done iteratively in a series of workshops. In the end, the model was considered adequate with regard to the influencing factors listed above. The behavior exhibited by the model was deemed correct by both experts and management at the company.

One important measure was the overall processing time of the request triggering the process. Figure 5-6 shows the processing time and its distribution. The workload of the organizational units involved was analyzed in the same way. These results were presented to the experts of the process to verify the quality of the simulation.

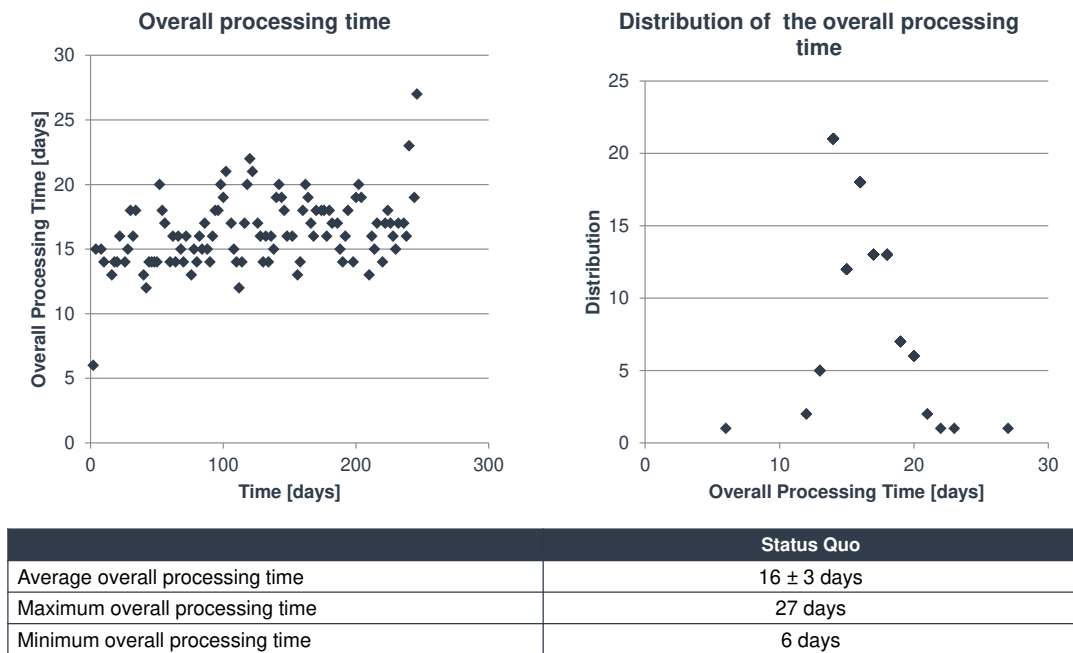


Figure 5-6 Status Quo – Overall processing time and its distribution

Additionally, the sufficient representation of reality by the System Dynamics model was verified by conducting model tests according to Sterman (2000). Within this use case, the tests boundary adequacy, extreme conditions, behavior reproduction and sensitivity analysis were conducted. During the test “boundary adequacy” the triggered incoming requests as the only element connected to the system boundary were scrutinized by different triggering forms. During the test “extreme conditions” several parameters of the model were set to extreme values, i.e. zero or 100% and the effects on the model were questioned for adequacy. The test “behavior reproduction” shall validate if the model reproduces the behavior of interest in the system Sterman (2000). The verification of the simulation results by presenting it to process experts as previously described can be seen as test “behavior reproduction”. During test “sensitivity analysis” several parameters of the model were varied over their plausible range and the results were analyzed for adequacy. The random part of the triggering input function for the first process step, the flow probabilities and the frequencies DBA, DBB and post-processing were examined in detail.

5.7 Step 4: Simulative analysis

The simulative analysis had two parts. Within the first part it was examined if the developed simulation model increases the understanding of the complexity of the engineering design process with special focus on the critical elements of the process. The second part focused on understanding the complexity of the overall process, which was the basis for the subsequent scenario analysis as it focused on the questions initially stated by the company.

The structural matrix view alone allows identifying feedback loops in sequences. As these loops create uncertainty or even lead to uncontrollable oscillation within the system, they are seen as critical elements. However, the results stay on a qualitative level. It was therefore monitored with the developed simulation model:

- if critical elements (identified by the metrics of Kreimeyer & Lindemann (2011)) can also be identified as critical within the simulation
- if the simulation offers a more detailed insight into the elements' criticalities.

Table 5-3 shows the process step elements, their criticality and the results of the simulation. High values are indicated in red. Two metrics of the simulation are used: the *maximum amounts of requests waiting to be processed* within this process step and the *overall amount of requests being processed* at this step over time. It can be seen that the two most critical elements according to structural metrics were also identified as most critical by the simulation (*discuss requests with DBA* and *implement decisions into master data set*). The first process step *check incoming requests*, which has the highest amount of overall requests passed, can be neglected for this analysis. The third most critical element *discuss requests with DBB* is not critical according to the simulation. This is due to the fact that the structural metric does not incorporate quantitative data, but each element is seen as equal. The quantitative data shows that this element does not seem critical, as it is not passed by a high amount of elements, which can be explained by the modeled flow probabilities.

Table 5-3 Critical elements of the process according to structural metrics and simulation measures

		Check incoming request	Prepare request for DBA	Prepare request for DBB	Discuss requests w/in DBA	Discuss requests w/in DBB	Handle queries of DBA w/in divisions	Escalate DBA queries to h. m.	Handle queries of DBB w/in divisions	Implement decisions into catalog	Implement decisions into m. d.	Validate master dataset	Solve contra-dictions	Release results
Structural complexity	Criticality	0	1	1	9	6	2	1	2	4	8	2	3	0
Simulation of dynamic complexity	Max amounts of req. waiting to be processed	6	5	1	13	4	6	7	6	4	12	6	1	-
	Overall am. of req. processed	248	198	50	244	54	73	49	38	50	218	109	22	-

The industry partner was also interested in identifying bottlenecks and cost drivers within the process which is shown in Table 5-4. In the specific context of the use case, bottlenecks were elements that delay the process. Usually, dependency models are only used to identify potential bottlenecks qualitatively by use of triangulation and assessing the amount and length of feedback loops. Within this use case, bottlenecks were analyzed quantitatively by the simulation. Therefore the amounts of elements passing a process step were multiplied with the average processing time of the particular step. Figure 12 shows the results in the middle part.

In addition to critical elements and bottlenecks, cost drivers were identified. For this purpose, the effort for each process step over the overall process time was calculated. Additionally, this measure was normalized based on the simulation results to show which effort is necessary for each step in percent of the overall effort. In the lower part, Table 5-4 shows the effort for each process step.

Table 5-4 Bottlenecks and cost drivers of the process according to the simulation

		Check incoming request	Prepare request for DBA	Prepare request for DBB	Discuss requests w/in DBA	Discuss requests w/in DBB	Handle queries of DBA w/in divisions	Escalate DBA queries to h. m.	Handle queries of DBB w/in divisions	Implement decisions into catalog	Implement decisions into m. d.	Validate master dataset	Solve contra-dictions	Release results
Identification of bottlenecks	Average processing time (in days)	1 ± 0	1 ± 0	1 ± 0	3 ± 2	5 ± 3	9 ± 0	13 ± 6	19 ± 0	2 ± 1	2 ± 1	1 ± 0	1 ± 0	-
	Characteristic number (processing time * am. of. elements)	241	193	48	482	234	1195	1081	2657	156	333	194	39	-
Identification of cost drivers	Overall effort [h]	20,7	41,3	87,0	43,5	48,7	24,3	6,3	1,7	36,2	196,0	-	-	
	Overall effort [%]	4,1 %	8,2 %	17,2 %	8,6 %	9,6 %	4,8 %	1,2 %	0,3 %	7,2 %	38,8 %	-	-	

To analyze the impact of the identified influencing factors their maximum and minimum values were defined in close cooperation with the industry partner. It was noted by the industry partner that the influencing factors had a varying degree of adjustability. This is why they were categorized into three categories for the further procedure: easy to affect, rather difficult to affect, very difficult to affect.

Consequently, the parameter values of the influencing factors were systematically varied within the simulation to analyze their impact. To assess the impact important measures for the process were defined. These were: overall processing time, workload for the organizational units involved and amount of unsolved requests. Figure 5-7 shows the results of the analysis as

presented to the process owners. The columns show the influencing factors and the rows present the impact measures for the process. Figure 5-7 can be read as follows: If the influencing factor in the corresponding column is varied in the given direction, this will impact the respective measure in the way indicated.

Influencing factors \ Impact on measures	Frequency DBA	Frequency DBB	Frequency post-processing	Quality Gate	Probability of iteration (DBA)	Probability of escalation (DBA)	Probability of iteration (DBB)	Maximum workload (DBA)	Maximum workload (DBB)
Overall processing time	↗	↗	↗	↘	↘	↘	↘	↗	↗
Workload for DBA	↗	↘	↘	↘	↘	↘	↘	↘	↘
Workload for DBB	↘	↗	↘	↘	↘	↘	↘	↘	↘
Workload for Org_A	↘	↘	↘	↘	↘	↘	↘	↘	↘
Unsolved requests	↗	↗	↗	↘	↘	↘	↘	↗	↗

↗ increasing relation
 ↘ neutral relation
 ↘ decreasing relation

Figure 5-7 Results of the SD analysis presented to the process owners

5.8 Step 5: Design application

Even though the following scenario analysis can be assigned to the previous step 4: Simulative analysis, the scenarios and its results are assigned to step 5: Design application here. The main purpose of the design application is the preparation of the generated information and sufficient documentation for future use. As the presentation of the scenario analysis of the use case is result oriented, the results of the scenario analysis, as part of the complexity analysis report, are included in step 5: Design application.

The complete complexity analysis report is not presented here because it comprises information which was previously stated within the description of the use case. The description of implications on the engineering design process structure and necessary actions can be found in the subsequent evaluation section as they were identified with the industry partner during feedback sessions.

To identify optimization potentials by scenario analysis, the results of Figure 5-7 were formatted the reverse way: If a measure is to be changed in a particular direction, how do the parameter values of the influencing factors have to be specified. In other words: how are the influencing factors influenced by the measures. See Figure 5-8.

Influencing factors / Optimization of measures	Frequency DBA	Frequency DBB	Frequency post-processing	Quality Gate	Probability of iteration (DBA)	Probability of escalation (DBA)	Probability of iteration (DBB)	Maximum workload (DBA)	Maximum workload (DBB)
Overall processing time	↘	↘	↘	↗	↘	↘	↘	↗	↗
Workload for DBA	↘	↗	↗	↗	↗	↗	↗	↘	↗
Workload for DBB	↗	↘	↗	↗	↗	↗	↗	↗	↘
Workload for Org_A	↗	↗	↗	↗	↘	↘	↘	↗	↗
Unsolved requests	↘	↘	↘	↗	↘	↘	↘	↗	↗

↗ increasing relation
 ↔ neutral relation
 ↘ decreasing relation

Figure 5-8 Concentrated results with a focus on optimization potentials

Based on the insights from Figure 5-8, six scenarios with differing optimization goals were defined. The scenarios ranged from decreasing the overall processing time or decreasing the workloads of the organizational units to multi-criteria optimization (decreasing overall processing time without increasing workload too much) to scenarios like a multi-project situation.

Figure 5-9 shows the scenario analysis for the first four scenarios. For each scenario it indicates how the influencing factors have to be changed to fulfill the scenario, and the particular assumption for each influencing factor is given for each scenario. The amount of change implemented into the simulation of each influencing factor depended on their previous categorization as “easy to affect”, “rather difficult to affect”, “very difficult to affect”. The more easily it was assumed to influence an influencing factor, the higher was the amount of possible change of its characteristics. The right side shows how the impact measures would be affected considering the assumptions implemented within each scenario.

Optimization goal of the scenario	Easy to affect							Rather difficult to affect					Very difficult to affect					Measures/ Results				
	Frequency DBA	Frequency DBB	Frequency post-processing	Quality Gate	Probability of iteration (DBA)	Probability of escalation (DBA)	Probability of iteration (DBB)	Maximum workload (DBA)	Maximum workload (DBB)	Overall processing time	Workload for DBA	Workload for DBB	Workload for Org_A	Unsolved requests	Workload for DBA	Workload for DBB	Workload for Org_A	Unsolved requests				
Status Quo	5 days	10 days	5 days	0%	30%	20%	70%	15 requests	5 requests	16 ± 3 days	4 ± 3 requests	2 ± 1 requests	1,73 h / Tag (=100 %)	13 ± 8 requests	4 ± 3 requests	2 ± 1 requests	1,73 h / Tag (=100 %)	13 ± 8 requests				
Scenario 1	5 days ↑	5 days ↘	5 days ↑	0% ↑	30% ↑	20% ↑	70% ↑	15 requests ↑	5 requests ↑	-13,5% ↘	+/- 0,0% ↗	-22,2% ↗	+8,3% ↗	-13,6% ↗	+/- 0,0% ↗	-22,2% ↗	+8,3% ↗	-13,6% ↗				
Scenario 2	10 days ↘	20 days ↘	10 days ↘	0% ↑	30% ↑	20% ↑	70% ↑	15 requests ↑	5 requests ↑	+88,7% ↗	+100,2% ↗	+47,5% ↗	-12,7% ↗	+95,3% ↗	+100,2% ↗	+47,5% ↗	-12,7% ↗	+95,3% ↗				
Scenario 3	10 days ↘	20 days ↘	10 days ↘	10% ↘	25% ↘	15% ↘	60% ↘	15 requests ↑	5 requests ↑	+60,9% ↗	+75,3% ↗	+37,1% ↗	-19,3% ↗	+47,4% ↗	+75,3% ↗	+37,1% ↗	-19,3% ↗	+47,4% ↗				
Scenario 4 (Scenario 1 + 3)	5 days ↑	5 days ↘	5 days ↑	10% ↘	25% ↘	15% ↘	60% ↘	15 requests ↑	5 requests ↑	-21,9% ↘	-12,4% ↘	-24,8% ↘	+1,8% ↗	-30,4% ↘	-12,4% ↘	-24,8% ↘	+1,8% ↗	-30,4% ↘				

Figure 5-9 Scenario analysis for the first four scenarios

5.9 Evaluation of results and feedback from the industry partner

The aim of the evaluation was to find out if the initial research question can be confirmed:

Can the framework for structure-based System Dynamics Analysis be applied in an industrial context and does it successfully serve as a tool to analyze engineering design processes and deduce improvement actions?

The evaluation was based on the experience of the process experts involved throughout the case study and their assessment of the accuracy and comprehensibility of the use case results. The evaluation was conducted through face-to-face interviews supported by a structured questionnaire with six process experts involved in the daily process routine at the industry partner. After statistical questions and a presentation of the use case results the participants were asked the questions presented in Table 5-5.

Table 5-5 Questions of the evaluation with regard to the results of the use case

No.	Question
Q1	Do you think that the MDM was useful to reveal the dependencies of the process?
Q2	Do you think that your understanding of the dependencies of the process has improved because of the MDM?
Q3	Do you think that your understanding of the dependencies of the process has improved because of the System Dynamics Simulation?
Q4	If your understanding has changed: in what way?
Q5	Has your understanding regarding the influencing factors (frequencies, maximum workloads, etc.) changed?
Q6	Has your understanding of critical elements, bottlenecks and cost drivers of the process changed?
Q7	Did you expect further knowledge which has not been provided by the System Dynamics Simulation?
Q8	Do you think that the results of the System Dynamics Simulation could be useful to optimize the process in industry?
Q9	Do you think that the combined use of matrix-based and System Dynamics approaches is appropriate for optimizing engineering design processes?

In the following a summary of the answers received is presented; the reference to the relevant question is added in brackets.

Overall, the MDM was seen as a useful tool to illustrate the dependencies of the process (Q1), but it was noted that flowcharts are generally better suited. The majority of the participants had the feeling that they better understood the dependencies of the process through the MDM (Q2). However, the System Dynamics simulation improved the perceived understanding of the process for most participants (Q3). The transparent presentation of the sequence, the influencing factors (control variables) and the quantitative data most significantly helped to improve the understanding (Q4). For most evaluation participants the understanding of the influencing factors acting on the process had changed (Q5). Also, the understanding of bottlenecks, cost drivers and cycle-barriers of the process could be enhanced. Only one participant stated that his existing knowledge had only been confirmed by the simulation (Q6). The participants were also asked about insights they had expected from the simulations, but which in the end were not provided (Q7). The request for a modified system boundary was stated several times, which would offer the possibility to also assess the workload for other organizational units not incorporated in the use case. It was also noted that the particular personalities of the people conducting a process step had an impact on the efficiency and

effectiveness of its execution. Another issue was the influence of the quality of incoming queries and the influence of preparation for particular process steps. Here, a scenario with varying quality (input and preparation) was requested.

The evaluation participants agreed that the resulting optimization scenarios of the System Dynamics simulation were useful for the company (Q8).

The combined use of matrix-based and System Dynamics approaches was rated as beneficial by all participants for the purpose of optimizing the engineering design process (Q9).

6. General points of discussion

Six aspects shall be addressed in this section: correctness, completeness, consistency and clearness of the framework as well as its relevance for academia and industry. The discussion of these aspects is based on the experiences gathered from the case studies and the use case including the evaluation and feedback from the industry partner, as well as insights from literature.

To achieve this thesis's objective of supporting the early phase of engineering design processes, the framework for structure-based System Dynamics Analysis of Engineering Design Processes was developed. It answers the initial research question of how to use structural models in form of MDMs as a basis to construct System Dynamics models.

This discussion section focuses on the six aspects of correctness, completeness, consistency, and clearness of the framework as well as its relevance for academia and industry:

Correctness: The framework allows modeling the sequence of an engineering design process with respect to its structural and dynamic complexity. Within the use case, the derived behavioral models were able to correctly model the dynamics of the engineering design process. The derived simulation results showed the behaviors expected by the process experts, and enabled scenario analysis. This allowed for a first reasonable assessment of potential scenarios. The boundaries of the scenario development, however, were not examined in this work.

Completeness: Within the cases examined, all desired artefacts could be modeled. Thus, no constraints were identified within the chosen system boundaries. Although everything could be modeled, it should be further identified what needs to be modeled with respect to the purpose of the analysis. Currently the framework is focused on combining the structural and dynamic complexity dimensions rather than offering a complete modeling framework itself. However, the introduction of an explicit "step 0" for defining the goals and scope of the analysis and the basic modeling concept may be helpful to complete the framework from the modeling perspective. Specific methods such as the goal question metric paradigm (Basili, 1992) may support this step.

Consistency: A continuous multi-step construction and usage of the models is possible. The joint construction of the model by various stakeholders, in particular, may add additional benefit, as a common understanding of the process can be generated. The structural and behavioral models are consistent to each other, as the behavioral model is derived from the structural model. This is in line with literature, which states that system structure is the main driver of system behavior.

Clearness: The framework identifies key enablers and their influence on engineering design processes on two levels: on the structural level (Who does what? What is the process sequence?) and on the behavioral level (What influences the engineering design process and how strongly?). The visualizations of the models can be quickly understood and can be also used for management presentations. However, the behavior itself is currently not visualized clearly enough. Stocks and flows representations are not very suitable for management

presentations. However, it is often at the management that decisions based on these kinds of analyses need to be taken.

Relevance for research and academia: This framework combines the application of structural matrix-based models and behavioral System Dynamics models by enabling the transformation of matrix-based models into System Dynamics. Thus, it offers an approach to better understand the correlation of system structure and system behavior, which is known in literature, but rarely understood. In particular, the framework allows to create a conceptual model space for the joint consideration of the dimensions of structural and dynamic complexity. This model space can be used for experimentation to support the early phase of engineering design processes. The relevance and applicability of the framework were shown. The presented framework may be generalizable for business processes. The boundaries for the definition of scenarios are currently not defined and need to be identified.

Relevance for industry: Existing process models can be used as an input for the suggested framework. Based on the specific model purpose, these models may need to be extended. The System Dynamics models provide a simulation environment for the purpose of process analysis and optimization and may thus, reduce the necessity of trial and error process management. The effort to conduct the structure-based System Dynamics analysis is relatively high, while at the same time the informative value seems to be high as well. The framework may be especially applicable to detailed process analysis and incremental improvements.

7. Conclusion and outlook

The previous chapters addressed the relevant state of the art, the case studies which led to the iterative development of the framework for structure-based System Dynamics Analysis – addressing the combined consideration of the static-structural and the dynamic-behavioral view on engineering design processes – as well as its procedure, evaluation and discussion. This chapter aims to provide a comprehensive overview and reflect about the work in this thesis from an industrial and academic perspective. Moreover, opportunities for further research are indicated.

7.1 Conclusion

Engineering design processes are not repeatable and inherently unpredictable in that completing an activity may result in a less complete state (Wynn et al., 2003). From a systems theory perspective, engineering design processes can thus be seen as complex systems comprising a particular structural as well as dynamic complexity.

Both, structural and dynamic complexity have been object to previous research. Structural and dynamic complexity are interrelated. This interrelatedness can, for example, be seen as structurally complex systems are usually also behaviorally (dynamically) complex (De Weck et al., 2011). Various researchers such as Baldwin & Clark (2000), Lindemann et al. (2005), Maurer (2007), Sharman & Yassine (2004) claim that system behavior can be estimated based on the structure of the system.

However, the interrelation of structural and dynamic complexity is not well understood and is still an object of research (Geraldi et al., 2011; Karniel & Reich, 2013; Rouse, 2007). When formulating his research needs for complex engineered, organizational and natural systems, Rouse (2007) explicitly demands conceptual frameworks that understand the interrelation of system structure and behavior as the first research need. From an industry perspective, understanding this interrelation can be a competitive lever for decreasing development cost and time. (Kreimeyer & Lindemann, 2011).

A systematic methodological framework – the framework for structure-based System Dynamics Analysis - is presented in this thesis to fulfill that need. It allows for the creation of a conceptual model space for the joint consideration of the dimensions of structural and dynamic complexity for supporting the early phase of engineering design processes. This model space can be used for experimentation to:

- understand the intrinsic complexity with its dimensions (i.e. structural and dynamic) and their interactions
- decrease the perceived complicatedness
- uncover knowable unknown unknowns.

Through the application of the framework, structural information of engineering design processes can be used as a basis to derive insights on the process behavior and vice versa. From the System Dynamics perspective, the framework serves the need for standard System

Dynamics structures to offer an accessible and reliable source of guidance for developing System Dynamics models (Warren, 2014). In contrast to existing Agent-based and Discrete-event simulations on top of structural models, the System Dynamics approach of the framework especially emphasizes a top-down view on the underlying process to support system understanding, decrease complicatedness and uncover risks. Particularly for planning and management activities during the early phase of engineering design processes the top-down perspective seems beneficial.

The framework was developed based on the state of the art and the insights of case studies. The initial academic case as well as the three most important and insightful case studies for the development of the framework for structure-based System Dynamics Analysis of Engineering Design Processes are documented in this thesis: the first academic case study examined the engineering design process of an e-bike sharing system. The three industrial case studies covered a line-process optimization, a benchmarking process analysis and the analysis of a construction process. From case to case, the framework was further developed and iteratively refined to answer each aspect of the initial research question as stated in the introductory chapter.

The generic framework is suitable for engineering design processes and may be tailored to specific boundary conditions and needs. It combines the views on engineering design processes from the structural and dynamic complexity dimension and offers special support for the transformation between these views.

The framework for structure-based System Dynamics Analysis consists of four steps: the starting point is the *MDM modeling*, which unfolds the structural complexity of the engineering design process. The step is followed by the *transformation and quantification* of the MDM model into a System Dynamics model. It is the essential step for combining the dimensions of structural and dynamic complexity. The transformation and quantification is split into two sub steps – *qualitative System Dynamics modeling* and *quantitative System Dynamics modeling*. The sub-step of qualitative System Dynamics modeling focuses on the transformation of the structural model into a System Dynamics model (the qualitative model). Within the sub-step of quantitative System Dynamics modeling the model is populated by equations and values for each component of the model to create a System Dynamics simulation model (the quantitative model). Before the System Dynamics model can be used for analysis, the step of *testing of the System Dynamics model* reviews if the developed model sufficiently represents reality. After successful testing the step of *simulative analysis* allows to derive insights on the behavior of the engineering design process. The *design application* makes use of the results of the system analysis in order to provide insights into the considered complexity challenge.

The framework supports all steps by providing various additional tools and methods. Among them are a MDM metamodel for system definition, a System Dynamics modeling classification of engineering design processes and heuristics for quantification. The framework allows creating a conceptual model space that lends itself to experimentation. The step of *simulative analysis* highlights three different types of analysis through experimentation: *Understanding the complexity: Uncovering risks* supports risk management by uncovering knowable unknown unknowns. *Future conditions: Scenario analysis* takes a “what if?” perspective and simulates

the effects of potential future conditions. *Performance of structural designs: Benchmarking* assesses the performance of different structural designs of an engineering design process.

The elaborated framework has been validated within a use case at a large German manufacturer of commercial vehicles. The results of the use case as well as the use of the framework as such were evaluated by face-to-face interviews with process experts, who stated that the resulting optimization scenarios from the structure-based System Dynamics Analysis were useful for the company and that the framework offered an overall benefit for the purpose of optimizing their engineering design processes. Their feedback included the fact that even though the structural model itself had already increased their system understanding, the System Dynamics analysis had further improved the actual process understanding of most participants. Furthermore, for most of the evaluated participants the understanding of the influencing factors acting on the process had changed due to the combined application of structural and behavioral models. Additionally, their understanding of the process's bottlenecks, cost drivers and cycle-barriers was enhanced.

7.2 Outlook

During the evaluation use case, feedback from process experts revealed some aspects that have to be addressed in follow-up research. Some experts asked for a modified system boundary that would offer the possibility to also assess the workload for other organizational units not incorporated in the use case. Other aspects raised, such as person-specific impacts like the performance of one particular individual and aspects influencing the quality of the process results, also point out the current simulation's lack of system boundary definition. A categorization of general industrial conditions based on their occurring dynamics for which the framework can be applied may be useful.

For the behavioral analysis, System Dynamics proves its suitability due to its adaptable degree of abstraction, its management perspective and the preliminary work within the SD community on engineering design processes by Cooper (1980), Ford & Sterman (1998a) and Le (2013). However, there exist several other modeling approaches for behavior, such as agent-based models, discrete event simulation or Petri nets. From research purposes, it needs to be clarified for which underlying challenges other behavioral modeling approaches are more suitable. One approach will not work for all problems (Bahill & Szidarovszky, 2009). Building on the mentioned categorization in the previous paragraph, best suiting modeling approaches for the particular conditions can be identified. It may also be insightful to analyze for which of the causes of iteration (poor activity sequencing, missing activities, poor communication, input changes and mistakes according to Lévárdy & Browning (2009)) which modeling approach and also on a more detailed level which rework cycle constructs should be used.

For the use case it was sufficient to model the process subset on a rather abstract level within the structural model. Additionally, it may often be possible to build the model upon existing process documentation. Based on the feedback from the process experts and the experience from the previous case studies, the cost-benefit view of the approach is promising as long as the amount of structural elements and dependencies is reasonable. 50 structural elements theoretically allow for up to 2,500 dependencies. An automated transformation may significantly reduce the modeling effort for cases with higher numbers of elements.

Referring to the initial research question, the framework shall be used for prescriptive process analysis. However, the dilemma of how to get the information for modeling the engineering design process in the early phase remains unsolved. This especially applies for the necessary quantitative information for the System Dynamics simulation model. As structural process information is usually available early in the engineering design process, its usage as basis to build the System Dynamics model is helpful, but does not solve the challenge completely.

8. References

- Adam, D. (Ed.). (1998). *Komplexitätsmanagement*. Wiesbaden: Gabler.
- Bahill, A. T., & Szidarovszky, F. (2009). Comparison of dynamic system modeling methods. *Systems Engineering*, 12(3), 183–200. <http://doi.org/10.1002/sys.20118>
- Bajaj, A., Kekre, S., & Srinivasan, K. (2004). Managing NPD: Cost and Schedule Performance in Design and Manufacturing. *Management Science*, 50(4), 527–536. <http://doi.org/10.1287/mnsc.1030.0177>
- Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: The power of modularity* (Vol. 1). MIT press.
- Bartolomei, J. E., Hastings, D. E., de Neufville, R., & Rhodes, D. H. (2012). Engineering Systems Multiple-Domain Matrix: An organizing framework for modeling large-scale complex systems. *Systems Engineering*, 15(1), 41–61. <http://doi.org/10.1002/sys.20193>
- Basili, V. R. (1992). *Software modeling and measurement: the Goal/Question/Metric paradigm*. Technical Report. CS-TR-2956. University of Maryland.
- Baumberger, G. C. (2007). *Methoden zur kundenspezifischen Produktdefinition bei individualisierten Produkten*. Lehrstuhl für Produktentwicklung. Dr. Hut. <http://doi.org/978-3-89963-660-4>
- Beck, K. (1999). *Extreme programming explained: embrace change*. Addison-Wesley.
- Becker, J., Kugeler, M., & Rosemann, M. (2003). *Process management: a guide for the design of business processes*. Berlin: Springer Science & Business Media.
- Belhe, U., & Kusiak, A. (1996). Modeling Relationships Among Design Activities. *Journal of Mechanical Design*, 118(4), 454. <http://doi.org/10.1115/1.2826912>
- Bhuiyan, N., Gerwin, D., & Thomson, V. (2004). Simulation of the new product development process for performance improvement. *Management Science*, 50(12), 1690–1703.
- Bias, R. G., & Mayhew, D. J. (2005). *Cost-justifying usability: an update for an Internet age* (Vol. 2). Amsterdam; Boston: Morgan Kaufman.
- Biedermann, W., Diepold, K. J., Lindemann, U., & Lohmann, B. (2012). Delimiting Structural and Dynamical System Analysis in Engineering Management. In D. Marjanovic, M. Storga, N. Pavkovic, & N. Bojetic (Eds.), *Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia* (pp. 1649–1658). Dubrovnik, Croatia: Design Society.
- Biemans, F. P. M., Lankhorst, M. M., Teeuw, W. B., & van de Wetering, R. G. (2001). Dealing with the Complexity of Business Systems Architecting. *Systems Engineering*, 4(2), 118–133. <http://doi.org/10.1002/sys.1010>
- Black, L. J., & Repenning, N. P. (2001). Why firefighting is never enough: Preserving high-quality product development. *System Dynamics Review*, 17(1), 33–62. <http://doi.org/10.1002/sdr.205>
- Blessing, L. T. M. (1994). *A process-based approach to computer-supported engineering*

design. University of Twente, Enschede.

- Blessing, L. T. M., & Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. Media. London: Springer London. <http://doi.org/10.1007/978-1-84882-587-1>
- Boehm, B., & Turner, R. (2003). *Balancing agility and discipline*. Boston, MA, USA .
- Bood, R., & Postma, T. (1997). Strategic learning with scenarios. *European Management Journal*, 15(6), 633–647. [http://doi.org/10.1016/S0263-2373\(97\)00047-9](http://doi.org/10.1016/S0263-2373(97)00047-9)
- Borshchev, A., & Filippov, A. (2004). From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons , Techniques , Tools 1 . Simulation Modeling: Abstraction Levels , Major Paradigms. In *The 22nd International Conference of the System Dynamics Society* (p. 45). Oxford, England.
- Box, G. E. P., & Draper, N. N. R. (1987). *Empirical model-building and response surfaces*. New York: Wiley.
- Bradley, M. J., Rudolph, J. W., & Carroll, J. S. (2013). Dynamic modeling as a multidiscipline collaborative journey. *System Dynamics Review*, 29(1), 4–25. <http://doi.org/10.1002/sdr.1492>
- Brooks, F. P. (1975). *The Mythical Man-Month*. Boston and others: Addison-Wesley.
- Brooks, R. J., & Tobias, a. M. (1996). Choosing the best model: Level of detail, complexity, and model performance. *Mathematical and Computer Modelling*, 24(4), 1–14. [http://doi.org/10.1016/0895-7177\(96\)00103-3](http://doi.org/10.1016/0895-7177(96)00103-3)
- Browning, T. R. (1998). *Modeling and analyzing cost, schedule, and performance in complex system product development*. Massachusetts Institute of Technology.
- Browning, T. R. (2001). Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Transactions on Engineering Management*, 48(3), 292–306. <http://doi.org/10.1109/17.946528>
- Browning, T. R. (2002). Process integration using the design structure matrix. *Systems Engineering*, 5(3), 180–193. <http://doi.org/10.1002/sys.10023>
- Browning, T. R. (2009). The many views of a process: Toward a process architecture framework for product development processes. *Systems Engineering*, 12(1), 69–90. <http://doi.org/10.1002/sys.20109>
- Browning, T. R., & Eppinger, S. D. (2002). Modeling impacts of process architecture on cost and schedule risk in product development. *IEEE Transactions on Engineering Management*, 49(4), 428–442. <http://doi.org/10.1109/TEM.2002.806709>
- Browning, T. R., Fricke, E., & Negele, H. (2006). Key concepts in modeling product development processes. *Systems Engineering*, 9(2), 104–128. <http://doi.org/10.1002/sys.20047>
- Browning, T. R., & Ramasesh, R. V. (2007). A Survey of Activity Network-Based Process Models for Managing Product Development Projects. *Production and Operations Management*, 16(2), 217–240. <http://doi.org/10.1111/j.1937-5956.2007.tb00177.x>
- Browning, T. R., & Ramasesh, R. V. (2015). Reducing unwelcome surprises in project management. *MIT Sloan Management Review*.

- Browning, T. R., & Yassine, A. A. (2010). Resource-constrained multi-project scheduling: Priority rule performance revisited. *International Journal of Production Economics*, 126(2), 212–228. <http://doi.org/10.1016/j.ijpe.2010.03.009>
- Brunies, R., & Emir, Z. (2001). Calculating loss of productivity due to overtime using published charts—Fact or fiction. *The Revay Report*, 20(3), 1–7.
- Carrascosa, M., Eppinger, S. D., & Whitney, D. (1998). Using the Design Structure Matrix to estimate product development time. In *Proceedings of the ASME design engineering technical conferences* (pp. 1–10). Atlanta, Georgia.
- Carson, J. S. (2002). Model verification and validation. In *Proceedings of the 2002 Winter Simulation Conference*. San Diego, CA.
- Casti, J. L. (1979). *Connectivity, complexity, and catastrophe in large-scale systems*. Chichester: Wiley.
- Chandrasekaran, B. (1990). Design problem solving: A task analysis. *AI Magazine*, 11(4), 59.
- Cho, S.-H., & Eppinger, S. D. (2005). A simulation-based process model for managing complex design projects. *IEEE Transactions on Engineering Management*, 52(3), 316–328. <http://doi.org/10.1109/TEM.2005.850722>
- Clevenger, C. M., Haymaker, J. R., & Ehrich, A. (2012). Design exploration assessment methodology: testing the guidance of design processes. *Journal of Engineering Design*, 24(3), 1–20. <http://doi.org/10.1080/09544828.2012.698256>
- Cooper, K. G. (1980). Naval Ship Production: A Claim Settled and a Framework Built. *Interfaces*, 10(6), 20–36. <http://doi.org/10.1287/inte.10.6.20>
- Cooper, K. G., & Lee, G. (2009). Managing the Dynamics of Projects and Changes at Fluor. In *Proceedings of the International System Dynamics Conference* (pp. 1–27). Albuquerque, NM, USA.
- Cooper, K. G., Lyneis, J. M., & Bryant, B. J. (2002). Learning to learn, from past to future. *International Journal of Project Management*, 20(3), 213–219. [http://doi.org/10.1016/S0263-7863\(01\)00071-0](http://doi.org/10.1016/S0263-7863(01)00071-0)
- Cornelius, P., Van De Putte, A., & Romani, M. (2005). Three decades of scenario planning in Shell. *California Management Review*, 48(1), 92–109. <http://doi.org/http://dx.doi.org/10.2307/41166329>
- Cronemyr, P., Ronnback, A. O., & Eppinger, S. D. (2001). A decision support tool for predicting the impact of development. *Journal of Engineering Design*, 12(3), 177–199. <http://doi.org/10.1080/09544820110055547>
- D’Avino, G., Dondo, P., Lo Storto, C., & Zezza, V. (2005). Reducing ambiguity and uncertainty during new product development: A system dynamics based approach. In *Portland International Conference on Management of Engineering and Technology* (Vol. 2005, pp. 538–549). Portland, OR, USA: IEEE. <http://doi.org/10.1109/PICMET.2005.1509735>
- Danilovic, M., & Browning, T. R. (2004). A formal approach for domain mapping matrices (DMM) to complement design structure matrices (DSM). In *6th International Design Structure Matrix Conference, Cambridge, UK, 12 - 14 September 2004*. Cambridge, UK.

- De Neufville, R. (2004). Uncertainty Management for Engineering Systems Planning and Design. In *Massachusetts Institute of Technology, Engineering Systems Division, Internal Symposium* (p. 19). Boston, MA, USA.
- De Weck, O. L., Roos, D., Magee, C. L., Vest, C. M., & Cooper, C. M. (2011). *Engineering Systems: Meeting Human Needs in a Complex Technological World*. Cambridge, USA: MIT Press.
- Doloi, H. (2010). Benchmarking a new design management system using process simulation approach. *Construction Innovation*, 10(1), 42–59. <http://doi.org/10.1108/14714171011017563>
- Dougherty, D. (2001). Reimagining the Differentiation and Integration of Work for Sustained Product Innovation. *Organization Science*, 12(5), 612–631. <http://doi.org/10.1287/orsc.12.5.612.10096>
- Dove, R. (2002). *Response Ability: The Language, Structure, and Culture of the Agile Enterprise* (Vol. 1). New York, NY, USA.
- Duffy, A. H. B., & MacCallum, K. J. (1989). Computer representation of numerical expertise for preliminary ship design. *Marine Technology*, 26(4), 289–302.
- Eckert, C. M., & Clarkson, J. P. (2005). Introduction: The reality of design. In *Design process improvement* (pp. 1–29). Springer.
- Eckert, C. M., Clarkson, J. P., & Stacey, M. (2001). Information flow in engineering companies: problems and their causes. In *International Conference on Engineering Design, ICED01* (Vol. 1, pp. 43–50).
- Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *Academy of Management Review*, 14(4), 532–550. <http://doi.org/10.5465/AMR.1989.4308385>
- Eisenhardt, K. M., & Tabrizi, B. N. (1995). Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry. *Administrative Science Quarterly*, 40(1), 84–110. <http://doi.org/10.2307/2393701>
- Eppinger, S. D., Whintey, D. E., Smith, R. P., & Gebala, D. a. (1994). A Model-Based Method for Organizing Task in Product Development. *Research Engineering Design*, 6, 1–13.
- Evans, J. H. (1959). Basic design concepts. *Journal of the American Society for Naval Engineers*, 71(4), 671–678.
- Fielding, N. G., & Fielding, J. L. (1986). *Linking Data. Qualitative research methods, Vol.4*. Thousand Oaks, CA, USA: SAGE Publications, Inc. <http://doi.org/10.4135/9781412984775>
- Flurschein, C. H. (1977). *Engineering Design Interfaces: a management philosophy*. London, UK: The Design Council.
- Ford, D. N., & Sterman, J. D. (1998a). Dynamic modeling of product development processes. *System Dynamics Review*, 14(1), 31–68.
- Ford, D. N., & Sterman, J. D. (1998b). Expert knowledge elicitation to improve formal and mental models. *System Dynamics Review*, 14(4), 309–340. [http://doi.org/10.1002/\(SICI\)1099-1727\(199824\)14:4<309::AID-SDR154>3.0.CO;2-5](http://doi.org/10.1002/(SICI)1099-1727(199824)14:4<309::AID-SDR154>3.0.CO;2-5)

- Ford, D. N., & Serman, J. D. (2003). Overcoming the 90% Syndrome: Iteration Management in Concurrent Development Projects. *Concurrent Engineering*, 11(3), 177–186. <http://doi.org/10.1177/106329303038031>
- Forrester, J. W. (1958). Industrial Dynamics--A Major Breakthrough for Decision Makers. *Harvard Business Review*, 36(4), 37–66.
- Foulds, L. R. (2012). *Graph theory applications*. Springer Science & Business Media.
- Garcia, R. (2005). Uses of Agent-Based Modeling in Innovation/New Product Development Research. *Journal of Product Innovation Management*, 22(5), 380–398. <http://doi.org/10.1111/j.1540-5885.2005.00136.x>
- Gemino, A., & Wand, Y. (2004). A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering*, 9(4), 248–260. <http://doi.org/10.1007/s00766-004-0204-6>
- Geraldi, J. G., Maylor, H., & Williams, T. M. (2011). Now, let's make it really complex (complicated): A systematic review of the complexities of projects. *International Journal of Operations & Production Management*, 31(9), 966–990. <http://doi.org/10.1108/01443571111165848>
- Gilbert, N. (2008). *Agent-Based Models*. SAGE Publications (Vol. 153). <http://doi.org/10.4135/9781412983259>
- Grebici, K., Wynn, D. C., & Clarkson, J. P. (2008). Modelling the Relationship between Uncertainty Levels in Design Descriptions and Design Process Duration. In *IDMME - Virtual Concepts 2008* (pp. 1–12). Beijing, China.
- Griffin, A., & Page, A. (1993). An interim report on measuring product development success and failure. *Journal of Product Innovation Management*, 10(4), 291–308. [http://doi.org/10.1016/0737-6782\(93\)90072-X](http://doi.org/10.1016/0737-6782(93)90072-X)
- Gross, J. L., & Yellen, J. (2005). *Graph theory and its applications*. CRC press.
- Haberfellner, R., Weck, O. De, Fricke, E., & Vössner, S. (Eds.). (2012). *Systems Engineering: Grundlagen und Anwendung* (12th ed.). Orell Füssli.
- Haeckel, S. (1999). *Adaptive enterprise*. Harvard Business School Press. Boston, MA, USA.
- Hale, M. L. (1997). Leading change. *The Journal of Academic Librarianship*, 23, 56. [http://doi.org/10.1016/S0099-1333\(97\)90081-1](http://doi.org/10.1016/S0099-1333(97)90081-1)
- Harter, D. E., Krishnan, M. S., & Slaughter, S. A. (2000). Effects of process maturity on quality, cycle time, and effort in software product development. *Management Science*, 46(4), 451–466.
- Haslett, T., & Sankaran, S. (2009). Applying Multi-Methodological System Theory to Project Management. *Proceedings of the 53rd Annual Meeting of the ISSS*, 1–12.
- Herold, S., Karch, H., & Schmidt, H. (2004). Shiga toxin-encoding bacteriophages - Genomes in motion. *International Journal of Medical Microbiology*, 294(2), 115–121. <http://doi.org/10.1016/j.ijmm.2004.06.023>
- Highsmith, J. (2004). Agile Project Management: Creating Innovative Products. *Management*, 69, 277.

- Highsmith, J. a, & Highsmith, J. (2013). Adaptive Software Development: A Collaborative Approach to Managing Complex Systems, 392.
- Hilmola, O.-P., Helo, P., & Maunuksela, A. (2005). The economic nature of feedback loops in product development. *International Journal of Innovation and Learning*, 2(2), 197. <http://doi.org/10.1504/IJIL.2005.006299>
- Hoffrage, U., & Pohl, R. F. (2010). Research on hindsight bias: a rich past, a productive present, and a challenging future. *Memory (Hove, England)*, 11(4-5), 329–35. <http://doi.org/10.1080/09658210344000080>
- Hubka, V. (1984). *Theorie Technischer Systeme*. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Hykin, D. H. W., & Laming, L. C. (1975). Design case histories: report of a field study of design in the united kingdom engineering industry. *Proceedings of the Institution of Mechanical Engineers 1847-1982 (Vols 1-196)*, 189, 203–211. http://doi.org/10.1243/PIME_PROC_1975_189_027_02
- Investopedia. (2016). Anchoring. Retrieved January 1, 2016, from <http://www.investopedia.com/terms/a/anchoring.asp>
- Jarratt, T. a. W., Eckert, C., Weeks, R., & Clarkson, P. J. (2003). Environmental legislation as a driver of design. In *International Conference on Engineering Design, ICED03* (pp. 1–10).
- Jarrett, J. P. (2001). *Technology or methodology? An approach to designing better turbomachinery*. University of Cambridge.
- Jick, T. D. (1979). Mixing Qualitative and Quantitative Methods : Triangulation in Action. *Administrative Science Quarterly*, 24, 602–611. <http://doi.org/10.2307/2392366>
- Jing, N. N., & Yang, C. (2009). The interrelationship among quality planning, knowledge process and new product development performance. In *Industrial Engineering and Engineering Management, 2009. IE&EM'09* (pp. 1051–1055). IEEE.
- Joglekar, N. R., & Ford, D. N. (2005). Product development resource allocation with foresight. *European Journal of Operational Research*, 160(1), 72–87. <http://doi.org/10.1016/j.ejor.2003.06.021>
- Karniel, A., & Reich, Y. (2009). From DSM-based planning to design process simulation: A review of process scheme logic verification issues. *IEEE Transactions on Engineering Management*, 56(4), 636–649. <http://doi.org/10.1109/TEM.2009.2032032>
- Karniel, A., & Reich, Y. (2013). Multi-level modelling and simulation of new product development processes. *Journal of Engineering Design*, 24(3), 185–210. <http://doi.org/10.1080/09544828.2012.720015>
- Kasperek, D., Berger, S., Maisenbacher, S., Lindemann, U., & Maurer, M. S. (2015). Structure-based System Dynamics Analysis - A Case Study of Benchmarking Process Optimization. In *International Conference on Engineering Design, ICED15* (pp. 1–10). Milano, Italy.
- Kasperek, D., Bermond, L., Maisenbacher, S., Zaggl, M. A., Raasch, C., & Maurer, M. S. (2015). Structure-based System Dynamics Analysis – A Case Study of Line Process

- Optimization. In *9th Annual IEEE International Systems Conference, IEEE SysCon15*. Vancouver, Canada: © 2011 IEEE.
- Kasperek, D., Kohn, A., & Maurer, M. S. (2013). Identifying Uncertainties within Structural Complexity Management. In *19th International Conference on Engineering Design 2013, ICED13*. (pp. 1–10). Seoul, South Korea: Design Society.
- Kasperek, D., Lichtenberg, C., Maisenbacher, S., Hollauer, C., Omer, M., & Maurer, M. S. (2015). Structure-based System Dynamics Analysis of Engineering Design Processes - A case study. In *9th Annual IEEE International Systems Conference, IEEE SysCon15*. Vancouver, Canada: © 2011 IEEE.
- Kasperek, D., Lindinger, M., Maisenbacher, S., & Maurer, M. S. (2014). A structure - based System Dynamics Approach for Assessing Engineering Design Processes. In *International System Dynamics Conference*. Delft, The Netherlands.
- Kasperek, D., Maisenbacher, S., Kohn, A., Lindemann, U., & Maurer, M. S. (2015). Increasing the reproducibility of structural modelling. *Journal of Engineering Design*, 26(7-9), 259–281. <http://doi.org/10.1080/09544828.2015.1026883>
- Kasperek, D., Maisenbacher, S., & Maurer, M. S. (2015). Structure-based Compilation of System Dynamics Models for Assessing Engineering Design Process Behavior. *The Journal of Modern Project Management*, 3(2), 101–107.
- Kasperek, D., & Maurer, M. S. (2013). Coupling Structural Complexity Management and System Dynamics to represent the dynamic behavior of product development processes. In *7th Annual IEEE International Systems Conference, IEEE SysCon13* (pp. 414–419). Orlando, USA: © 2011 IEEE. <http://doi.org/10.1109/SysCon.2013.6549915>
- Kasperek, D., Schenk, D., Kreimeyer, M., Lindemann, U., & Maurer, M. S. (2016). Structure-based System Dynamics Analysis of Engineering Design Processes. *Systems Engineering (Submitted)*.
- Kernan, M. C., Bruning, N. S., & Miller-Guhde, L. (1994). Individual and Group Performance: Effects of Task Complexity and Information. *Human Performance*, 7(4), 273–289. http://doi.org/10.1007/978-1-4615-9239-6_1
- Kneuper, R. (2003). *CMMI*. Heidelberg: dpunkt-Verl.
- Kohn, A. (2014). *Entwicklung einer Wissensbasis für die Arbeit mit Produktmodellen*. Dr. Hut, München.
- Kortler, S., & Lindemann, U. (2011). A Meta Model of the Innovation Process to Support the Decision Making Process Using Structural Complexity Management. In T. J. Howard, K. Mougard, T. C. McAloone, & C. T. Hansen (Eds.), *International Conference on Engineering Design, ICED11*. Copenhagen, Denmark: Design Society.
- Koskela, L., & Howell, G. (2008). The underlying theory of project management is obsolete. *IEEE Engineering Management Review*, 36(2), 22–34. <http://doi.org/10.1109/EMR.2008.4534317>
- Krause, D., & Denzin, N. K. (1989). *The Research Act: A Theoretical Introduction to Sociological Methods*. Teaching Sociology (Vol. 17). <http://doi.org/10.2307/1318434>
- Kreimeyer, M. (2009). *A Structural Measurement System for Engineering Design Processes*.

Dr. Hut, München.

- Kreimeyer, M., Förg, A., & Lienkamp, M. (2014). Fostering Modular Kits in an industrial Brownfield Environment. In *Tools and Methods of Competitive Engineering (TMCE 2014)* (pp. 1–12). Budapest, Hungary.
- Kreimeyer, M., & Lindemann, U. (2011). *Complexity Metrics in Engineering Design*. Berlin, Heidelberg: Springer Berlin Heidelberg. <http://doi.org/10.1007/978-3-642-20963-5>
- Krishnan, V. (1996). Managing the simultaneous execution of coupled phases in concurrent product development. *IEEE Transactions on Engineering Management*, 43(2), 210–217.
- Krishnan, V., Eppinger, S. D., & Whitney, D. (1997). A Model-Based Framework to Overlap Product Development Activities. *Management Science*, 43(4), 437–451. <http://doi.org/10.1287/mnsc.43.4.437>
- Kvålseth, T. (1978). The effect of task complexity on the human learning function. *International Journal of Production Research*, 16(5), 427–435. <http://doi.org/10.1080/00207547808930033>
- Laverghetta, T., & Brown, A. (1999). Dynamics of Naval Ship Design: A Systems Approach. *Naval Engineers Journal*, 111(3), 307–323. <http://doi.org/10.1111/j.1559-3584.1999.tb01981.x>
- Law, A. (2014). *Simulation Modeling and Analysis* (5th ed.). New York, USA: McGraw-Hill Education.
- Le, H. N. (2013). *A Transformation-Based Model Integration Framework to Support Iteration Management in Engineering Design*. University of Cambridge.
- Le, H. N., Wynn, D. C., & Clarkson, J. P. (2012). Impacts of concurrency, iteration, design review, and problem complexity on design project lead time and error generation. *Concurrent Engineering*, 20(1), 55–67. <http://doi.org/10.1177/1063293X11435177>
- Lee, S. H., Peña-Mora, F., & Park, M. (2006). Dynamic planning and control methodology for strategic and operational construction project management. *Automation in Construction*, 15, 84–97. <http://doi.org/10.1016/j.autcon.2005.02.008>
- Lee, S., & Sung Lim, I. (2007). Degree of Overlapping Design Activities in Vehicle Development : A System Dynamics Approach. *Asian Journal on Quality*, 8(2), 128–144. <http://doi.org/10.1108/15982688200700019>
- Lessio, M. P. De. (2011). *Assessing the Complex Product Design Process Planning Activity*. University of Cambridge.
- Lévárdy, V., & Browning, T. R. (2009). An adaptive process model to support product development project management. *IEEE Transactions on Engineering Management*, 56(4), 600–620. <http://doi.org/10.1109/TEM.2009.2033144>
- Lichtenberg, C., Kasperek, D., Maisenbacher, S., & Maurer, M. S. (2013). Strukturbasierte Modellierung und Bewertung von Entwicklungsprozessen von Produkt-Service Systemen. In *Tag des Systems Engineering* (pp. 219–228). Stuttgart, Germany.
- Lilien, G. L., & Yoon, E. (1990). The Timing of Competitive Market Entry: An Exploratory Study of New Industrial Products. *Management Science*, 36(5), 568–585. <http://doi.org/10.1287/mnsc.36.5.568>

- Lin, J. (2006). Overlapping in Distributed Product Development. *Proceedings of the 24th International Conference of the System Dynamics Society*, 85–86.
- Lin, J., Chai, K. H., Wong, Y. S., & Brombacher, A. C. (2008). A dynamic model for managing overlapped iterative product development. *European Journal of Operational Research*, 185(1), 378–392. <http://doi.org/10.1016/j.ejor.2006.12.022>
- Lindemann, U. (2007). *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden*. Berlin, Heidelberg: Springer.
- Lindemann, U., Maurer, M. S., & Braun, T. E. (2009). *Structural Complexity Management. Structural Complexity Management: An Approach for the Field of Product Design*. Berlin, Germany: Springer. <http://doi.org/10.1007/978-3-540-87889-6>
- Lindemann, U., Maurer, M. S., & Kreimeyer, M. (2005). Intelligent strategies for structuring products. In J. Clarkson & M. Huhtala (Eds.), *Engineering Design – Theory and Practice* (pp. 106–115).
- Lisse, S. (2013). Applying system dynamics for outsourcing services in design-build projects. *Journal of Project, Program & Portfolio Management*, 4(2), 20–36.
- Liu, Y., & Liu, Z. (2010). Multi-objective product configuration involving new components under uncertainty. *Journal of Engineering Design*, 21(4), 473–494. <http://doi.org/10.1080/09544820802474663>
- Loch, C. H., DeMeyer, A., & Pich, M. T. (2006). *Managing the Unknown: A New Approach to Managing High Uncertainty and Risk in Projects: A New Approach to Managing High Uncertainty and Risk in Projects* (1st ed., Vol. 30). John Wiley & Sons.
- Loch, C. H., Solt, M. E., & Bailey, E. M. (2008). Diagnosing unforeseeable uncertainty in a new venture. *Journal of Product Innovation Management*, 25(1), 28–46. <http://doi.org/10.1111/j.1540-5885.2007.00281.x>
- Lorenz, M. (2008). *Handling of Strategic Uncertainties in Integrated Product Development. Lehrstuhl für Produktentwicklung*. Dr. Hut, München.
- Lyneis, J. M., Cooper, K. G., & Els, S. a. (2001). Strategic management of complex projects: A case study using system dynamics. *System Dynamics Review*, 17(3), 237–260. <http://doi.org/10.1002/sdr.213>
- Lyneis, J. M., & Ford, D. N. (2007). System dynamics applied to project management: A survey, assessment, and directions for future research. *System Dynamics Review*, 23(2-3), 157–189. <http://doi.org/10.1002/sdr.377>
- MacCormack, A., Verganti, R., & Iansiti, M. (2001). Developing Products on “Internet Time”: The Anatomy of a Flexible Development Process. *Management Science*, 47(1), 133–150. <http://doi.org/10.1287/mnsc.47.1.133.10663>
- Maier, A. M., Kreimeyer, M., Hepperle, C., Eckert, C. M., Lindemann, U., & Clarkson, J. P. (2008). Exploration of correlations between factors influencing communication in complex product development. *Concurrent Engineering*, 16(1), 37–59.
- Maier, A. M., & Störrle, H. (2011). What are the Characteristics of Engineering Design Processes? In *International Conference on Engineering Design, ICED11* (pp. 1–11).
- Maurer, M. S. (2007). *Structural Awareness in Complex Product Design. Lehrstuhl für*

Produktentwicklung. Dr. Hut, Munich, Germany.

- Maurer, M. S. (2011). Systematic Knowledge Transfer based on Knowledge Correlations. In *ICORD 11: Proceedings of the 3rd International Conference on Research into Design Engineering*. Bangalore, India.
- Maurer, M. S., Biedermann, W., Cole, M., D'Avanzo, J., & Dickmanns, D. (2009). Airport security: From single threat aspects to valid scenarios and risk assessment. In *1st Annual Global Conference on System and Enterpris (GCSE), December 2009, Washington DC, USA*. Washington, D.C.
- Maynard, D. C., & Hakel, M. D. (1997). Effects of Objective and Subjective Task Complexity on Performance. *Human Performance*, 10(4), 303–330. http://doi.org/10.1207/s15327043hup1004_1
- Meier, C., Browning, T. R., Yassine, A. A., & Walter, U. (2015). The cost of speed: Work policies for crashing and overlapping in product development projects. *IEEE Transactions on Engineering Management*, 62(2), 237–255. <http://doi.org/10.1109/TEM.2015.2411514>
- Meier, H., & Boßlau, M. (2012). Dynamic Business Models for Industrial Product-Service Systems. In Y. Shimomura & K. Kimita (Eds.), *The 30th International Conference of the System Dynamics Society* (pp. 1–20). Berlin: Springer. http://doi.org/10.1007/978-3-642-32847-3_30
- Mihm, J., Loch, C. H., & Huchzermeier, A. (2003). Problem–Solving Oscillations in Complex Engineering Projects. *Management Science*, 49(6), 733–750.
- Moon, Y. B. (2015). Simulation Modeling for Sustainability: A Review of the Literature. *Mechanical and Aerospace Engineering, Paper 15*, 1–36.
- Muench, D. (1994). The Sybase Development Framework. Oakland, CA, USA: Sybase Inc.
- Munoz Hernandez, J. I., Otegui Olaso, J. R., & Gutierrez, A. (2013). Technology Assessment in Software Development Projects Using a System Dynamics Approach: A Case of Application Frameworks. In *Engineering Management*. InTech. <http://doi.org/10.5772/54498>
- Murphy, S. A., & Kumar, V. (1997). The front end of new product development: a Canadian survey. *R&D Management*, 27(1), 5–15. <http://doi.org/10.1111/1467-9310.00038>
- Naeser, T. (2015). Centre for Advanced Laser Applications (CALA) - Laboratory for Extreme Photonics - LMU Munich. Retrieved November 18, 2015, from <http://www.lex-photonics.de/cala/>
- Nandakumar, P., Datar, S. M., & Akella, R. (1993). Models for measuring and accounting for cost of conformance quality. *Management Science*, 39(1), 1–16.
- Nasirzadeh, F., Khanzadi, M., Afshar, A., & Howick, S. (2013). Modeling quality management in construction projects. *International Journal of Civil Engineering*, 11(1), 14–22.
- Nepal, M. P., Park, M., & Son, B. (2006). Effects of Schedule Pressure on Construction Performance. *Journal of Construction Engineering and Management*, 132(2), 182–188. [http://doi.org/10.1061/\(ASCE\)0733-9364\(2006\)132:2\(182\)](http://doi.org/10.1061/(ASCE)0733-9364(2006)132:2(182))
- O'Donovan, B. D. (2004). *Modelling and Simulation of Engineering Design Processes*. University of Cambridge.

- Oberkampff, W. L., Helton, J. C., Joslyn, C. A., Wojtkiewicz, S. F., & Ferson, S. (2004). Challenge problems: uncertainty in system response given uncertain parameters. *Reliability Engineering & System Safety*, 85(1-3), 11–19. <http://doi.org/10.1016/j.ress.2004.03.002>
- Oehmen, J., Thuesen, C., Ruiz, P. P., & Geraldi, J. (2015). *Complexity Management for Projects, Programmes, and Portfolios: An Engineering Systems Perspective* (Project Management Institute, PMI. White paper).
- Oliva, R. (2003). Model calibration as a testing strategy for system dynamics models. *European Journal of Operational Research*, 151(3), 552–568. [http://doi.org/10.1016/S0377-2217\(02\)00622-7](http://doi.org/10.1016/S0377-2217(02)00622-7)
- Pahl, G., & Beitz, W. (1996). *Engineering design: a systematic approach*. London, UK: Springer.
- Pahl, G., Beitz, W., Feldhusen, J., & Grote, K. H. (2007). *Konstruktionslehre*. Berlin, Heidelberg, New York: Springer.
- Pahl, G., Beitz, W., Feldhusen, J., & Grothe, K. H. (2007). *Engineering design: a systematic approach*. (G. Pahl & K. Wallace, Eds.) (3. ed.). London, UK: Springer.
- Parvan, K., Rahmandad, H., & Haghani, A. (2013). Empirical Study of Design-Construction Feedbacks in Building Construction Projects. *31 St International Conference of the System Dynamics Society*, (c), 1–32.
- Paulson, D., & Wand, Y. (1992). An automated approach to information systems decomposition. *IEEE Transactions on Software Engineering*, 18(3), 174–189. <http://doi.org/10.1109/32.126767>
- Petz, A., Schneider, S., Duckwitz, S., & Schlick, C. M. (2014). Modeling and Simulation of Service Systems with Design Structure and Domain Mapping Matrices. In *International Dependency and Structure Modelling DSM Conference, DSM2014*. Paris, France.
- Pfahl, D., & Lebsanft, K. (1999). Integration of system dynamics modelling with descriptive process modelling and goal-oriented measurement. *Journal of Systems and Software*, 46(2-3), 135–150. [http://doi.org/10.1016/S0164-1212\(99\)00007-2](http://doi.org/10.1016/S0164-1212(99)00007-2)
- Piller, F. T., & Waringer, D. (1999). *Modularisierung in der Automobilindustrie - Neue Formen und Prinzipien*. Shaker.
- Pohlmeyer, A. E. (2011). *Identifying Attribute Importance in Early Product Development*. Technische Universität Berlin.
- Poppendieck, M., & Poppendieck, T. (2003). Lean software development: an agile toolkit. *Computer*, 36(8). <http://doi.org/10.1109/MC.2003.1220585>
- Powell, A., Mander, K., & Brown, D. (1999). Strategies for lifecycle concurrency and iteration – A system dynamics approach. *Journal of Systems and Software*, 46(2-3), 151–161. [http://doi.org/10.1016/S0164-1212\(99\)00008-4](http://doi.org/10.1016/S0164-1212(99)00008-4)
- Produktentwicklung. (2013). Semesterarbeit Lichtenberg: Strukturbasierte Modellierung und Bewertung eines Product-Service-Systems in der frühen Phase (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014a). Bachelorarbeit Bermond: Strukturbasierte System Dynamics-

- Modellierung zur Prozessoptimierung und -validierung (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014b). Semester Thesis Schmidt: A guideline for adapted System Dynamics modeling of rework cycles in engineering design processes (not published). Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014c). Semesterarbeit Berger: Analyse des MAN Benchmarking-Prozesses mittels strukturbasierter System Dynamics Modellierung (not published, confidential). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014d). Semesterarbeit Lichtenberg: Dynamische Verhaltenssimulation mittels System Dynamics (not published). München: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2015). Semester Thesis Schenk: Dynamic Structure Simulation - Strukturbasierte Simulation eines Entwicklungsprozesses der frühen Phase komplexer Großprojekte am Beispiel der Nutzfahrzeugindustrie (not published, confidential). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Project Management Institute Inc. (2013). A Guide to the Project Management Body of Knowledge - PMBOK Guide. *PMI Book*, 1–589.
- Rahmandad, H., & Al., E. (2005). Dynamics of platform-based product development. *Proceedings of the 23rd International System Dynamics Conference*.
- Rahmandad, H., & Hu, K. (2010). Modeling the rework cycle: capturing multiple defects per task. *System Dynamics Review*, 26(4), 291–315. <http://doi.org/10.1002/sdr.435>
- Rahmandad, H., & Sterman, J. D. (2008). Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation Models. *Management Science*, 54(5), 998–1014. <http://doi.org/10.1287/mnsc.1070.0787>
- Ramasesh, R. V., & Browning, T. R. (2014). A conceptual framework for tackling knowable unknown unknowns in project management. *Journal of Operations Management*, 32(4), 190–204. <http://doi.org/10.1016/j.jom.2014.03.003>
- Rannacher, A., Stranzenbach, R., Sturm, F., Mütze-Niewöhner, S., & M. Schlick, C. (2013). A System Dynamics Model for the Evaluation of the Productivity of Knowledge-intensive Services. *iBusiness*, 05(03), 55–58. <http://doi.org/10.4236/ib.2013.53B012>
- Rechtin, E. (1991). *Systems Architecting: Creating and Building Complex Systems*. *Systems architecting creating and building complex systems*. Englewood Cliffs, N.J: Prantice Hall.
- Reichelt, K., & Lyneis, J. M. (1999). The dynamics of project performance: benchmarking the drivers of cost and schedule overrun. *European Management Journal*, 17, 135–150. [http://doi.org/10.1016/S0263-2373\(98\)00073-5](http://doi.org/10.1016/S0263-2373(98)00073-5)
- Repenning, N. P. (2000). A dynamic model of resource allocation in multi-project research and development systems. *System Dynamics Review*, 16(3), 173–212. [http://doi.org/10.1002/1099-1727\(200023\)16:3<173::AID-SDR196>3.0.CO;2-E](http://doi.org/10.1002/1099-1727(200023)16:3<173::AID-SDR196>3.0.CO;2-E)
- Repenning, N. P., & Sterman, J. D. (2002). Capability Traps and Self-Confirming Attribution Errors in the Dynamics of Process Improvement. *Administrative Science Quarterly*, 47(2),

265. <http://doi.org/10.2307/3094806>
- Rittel, H. W. J., & Webber, M. M. (1984). Planning Problems are Wicked Problems. In *Developments in Design Methodology* (Vol. NA, pp. 134–144). John Wiley & Sons. <http://doi.org/10.1016/j.jacr.2013.08.013>
- Ritter, F. E., & Schooler, L. J. (2002). The learning curve. In *International Encyclopedia of the Social and Behavioral Sciences* (pp. 8602–8605). Amsterdam: Pergamon. <http://doi.org/10.1067/mhn.2002.121517>
- Roelofsen, J., Krehmer, H., Lindemann, U., & Meerkamm, H. (2008). Using the Design-Structure-Matrix for the Avoidance of Unnecessary Iterations. In *Proceedings of the 10th International DSM Conference* (pp. 209–218). Stockholm, Sweden.
- Roese, N. J., & Vohs, K. D. (2012). Hindsight Bias. *Perspectives on Psychological Science: A Journal of the Association for Psychological Science*, 7(5), 411–26. <http://doi.org/10.1177/1745691612454303>
- Rozenburg, N. F. M., & Eekels, J. (1995). *Product design: fundamentals and methods*. Chichester: Wiley.
- Rosenbaum, L. L., & Rosenbaum, W. B. (1971). Morale and productivity consequences of group leadership style, stress, and type of task. *Journal of Applied Psychology*, 55(4), 343–348. <http://doi.org/10.1037/h0031458>
- Roth, G. L., & Senge, P. M. (1996). From theory to practice: research territory, processes and structure at an organizational learning centre. *Journal of Organizational Change Management*, 9(1), 92–106. <http://doi.org/10.1108/09534819610107349>
- Rouse, W. B. (2007). Complex engineered, organizational and natural systems: Issues underlying the complexity of systems and fundamental research needed to address these issues. *Systems Engineering*, 10(3), 260–271. <http://doi.org/10.1002/sys.20076>
- Ruutu, S., Ylén, P., & Laine, M. (2011). Simulation of a distributed design project. *Proceedings of the 29th International System Dynamics Conference*.
- Safoutin, M. J. (2003). *A methodology for empirical measurement of iteration in engineering design processes*. University of Washington.
- Sarbacker, S. D., & Ishii, K. (1997). A Framework for Evaluating Risk in Innovative Product Development. In *Proceedings of the ASME DETC97* (pp. 1–10).
- Schein, E. H. (1990). *Process Consultation Volume 1 Its Role in Organizational Development*. FT Press.
- Schlick, C. M., Duckwitz, S., & Schneider, S. (2013). Project dynamics and emergent complexity. *Computational and Mathematical Organization Theory*, 19(4), 480–515. <http://doi.org/10.1007/s10588-012-9132-z>
- Schmenner, R. W., & Swink, M. L. (1998). On theory in operations management. *Journal of Operations Management*, 17(1), 97–113. [http://doi.org/10.1016/S0272-6963\(98\)00028-X](http://doi.org/10.1016/S0272-6963(98)00028-X)
- Schmidt, E., Kasperek, D., & Maurer, M. S. (2015). A Guideline for Adapted System Dynamics Modeling of Rework Cycles in Engineering Design Processes. In W. J. C. V. Richard Curran, Nel Wognum, Milton Borsato, Josip Stjepandić (Ed.), *Transdisciplinary Lifecycle Analysis of Systems* (Volume 2:, pp. 357 – 366). Delft, The Netherlands.

- <http://doi.org/10.3233/978-1-61499-544-9-357>
- Schoeneberg, K.-P. (2014). *Komplexitätsmanagement in Unternehmen: Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern*. Springer-Verlag.
- Schrader, S., Riggs, W. M., & Smith, R. P. (1993). Choice over uncertainty and ambiguity in technical problem solving. *Journal of Engineering and Technology Management*, 10(1-2), 73–99. [http://doi.org/10.1016/0923-4748\(93\)90059-R](http://doi.org/10.1016/0923-4748(93)90059-R)
- Schrieverhoff, P. (2014). *Valuation of Adaptability in System Architecture*. München: Dr. Hut.
- Schuh, G. (2005). *Produktkomplexität managen : Strategien, Methoden, Tools* (2., über). München [u.a.]: Hanser.
- Sharman, D. M., & Yassine, A. A. (2004). Characterizing complex product architectures. *Systems Engineering*, 7(1), 35–60. <http://doi.org/10.1002/sys.10056>
- Simon, H. A. (1962). The Architecture of Complexity. In *Proceedings of the American Philosophical Society* (Vol. 106, pp. 467–482).
- Smith, & Morrow, J. a. (1999). Product development process modeling. *Design Studies*, 20(3), 237–261. [http://doi.org/10.1016/S0142-694X\(98\)00018-0](http://doi.org/10.1016/S0142-694X(98)00018-0)
- Smith, G. R., Herbein, W. C., & Morris, R. C. (1999). Front-end innovation at AlliedSignal and Alcoa. *Research Technology Management*, 42(6), 15.
- Smith, R. P., & Eppinger, S. D. (1997). Identifying Controlling Features of Engineering Design Iteration. *Management Science*, 43(3), 276–293. <http://doi.org/10.1287/mnsc.43.3.276>
- Sosa, M. E., Eppinger, S. D., Pich, M., McKendrick, D. G., & Stout, S. K. (2002). Factors that influence technical communication in distributed product development: an empirical study in the telecommunications industry. *IEEE Transactions on Engineering Management*, 49(1), 45–58. <http://doi.org/10.1109/17.985747>
- Spur, G., & Eßer, G. (2013). *Produktionsinnovationen: Jahrbuch der inpro-Innovationsakademie 2012*. Carl Hanser Verlag GmbH & Company KG.
- Stachowiak, H. (1973). *Allgemeine Modelltheorie. Library* (Vol. formatik-S). Vienna: Springer. <http://doi.org/10.1007/978-3-7091-8327-4>
- Sterman, J. D. (2000). *Business dynamics: Systems thinking and modeling for a complex world*. Boston: Irwin/McGraw-Hill.
- Steward, D. V. (1981). Design Structure System: A Method for Managing the Design of Complex Systems. *IEEE Transactions on Engineering Management*, EM-28(3), 71–74. <http://doi.org/10.1109/TEM.1981.6448589>
- Storto, C. Lo, D'Avino, G., Dondo, P., & Zezza, V. (2008). Simulating information ambiguity during new product development: a forecasting model using system dynamics. *International Journal of Modelling, Identification and Control*, 3(1), 97. <http://doi.org/10.1504/IJMIC.2008.018190>
- Stuart, I., McCutcheon, D., Handfield, R., McLachlin, R., & Samson, D. (2002). Effective case research in operations management: A process perspective. *Journal of Operations Management*, 20(5), 419–433. [http://doi.org/10.1016/S0272-6963\(02\)00022-0](http://doi.org/10.1016/S0272-6963(02)00022-0)
- Suss, S., & Thomson, V. (2012). Optimal design processes under uncertainty and reciprocal

- dependency. *Journal of Engineering Design*, 23(10-11), 826–848. <http://doi.org/10.1080/09544828.2012.704546>
- Tatarczyk, B. (2009). *Organisatorische Gestaltung der frühen Phase des Innovationsprozesses: Konzeptionen, Methoden und Anwendung am Beispiel der Automobilindustrie*. Gabler Edition Wissenschaft. Wiesbaden: Gabler.
- Taylor, T., & Ford, D. N. (2006). Tipping point failure and robustness in single development projects. *System Dynamics Review*, 22(1), 51–71. <http://doi.org/10.1002/sdr.330>
- Technische Universität München. (2015). CRC 768 / SFB 768. Retrieved October 2, 2015, from <http://www.sfb768.tum.de/index.php?id=13&L=1>
- Terwiesch, C., & Loch, C. H. (1999). Measuring the Effectiveness of Overlapping Development Activities. *Management Science*, 45(4), 455–465. <http://doi.org/10.1287/mnsc.45.4.455>
- Thomas, H. R., & Napolitan, C. L. (1995). Quantitative Effects of Construction Changes on Labor Productivity. *Journal of Construction Engineering and Management*, 121(3), 290–296. [http://doi.org/10.1061/\(ASCE\)0733-9364\(1995\)121:3\(290\)](http://doi.org/10.1061/(ASCE)0733-9364(1995)121:3(290))
- Thomas, H. R., & Raynar, K. a. (1997). Schedule Overtime and Labor Productivity: Quantitative Analysis. *Journal of Construction Engineering and Management*, 123(2), 181–188.
- Tidstam, A., & Malmqvist, J. (2015). A systematic process for developing product configuration rules. *International Journal of Product Lifecycle Management*, 8(1), 46. <http://doi.org/10.1504/IJPLM.2015.068013>
- Tignor, W. (2004). System Engineering and System Dynamics models. In *Proceedings of the 22nd International Conference of the System Dynamics Society* (Vol. 34, p. 110). Oxford, England. <http://doi.org/10.1093/carcin/bgt388>
- Ullman, D. G., Dieterich, T. G., & Stauffer, L. A. (1988). A model of the mechanical design process based on empirical data. *AI EDAM*, 2(1), 33–52.
- Ulrich, K. T., & Eppinger, S. D. (2003). *Product design and development*. New York: McGraw Hill - Education.
- Vajna, S. (2005). Workflow for design. In *Design process improvement* (pp. 366–385). Springer.
- Van Der Aalst, W., & Van Hee, K. M. (2004). *Workflow management: models, methods, and systems*. Cambridge, USA: MIT press.
- VDI. (1994). *VDI 2247: Quality Management in Product Development*. Düsseldorf: VDI-Verlag.
- Visser, W. (2006). *The cognitive artifacts of designing*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates.
- von Neuman, J. (1947). The mathematician. *The Works of the Mind*, 1(1), 180–196.
- Warren, K. (2014). Standard Cases : Standard Structures : Standard Models . In *International Conference on System Dynamics* (pp. 1–20). Delft, The Netherlands.
- Wasson, C. (2006). *System analysis, design, and development: Concepts, principles, and*

- practices*. Hoboken, N.J: Wiley-Interscience.
- Wheelwright, S. C., & Clark, K. B. (1992). *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*. New York: NY. New York, NY: Free Press. <http://doi.org/citeulike-article-id:4107539>
- Wiebel, M., Eifler, T., Mathias, J., Kloberdanz, H., Bohn, A., & Birkhofer, H. (2013). Modellierung von Unsicherheit in der Produktentwicklung. In S. Jeschke, E.-M. Jakobs, & A. Dröge (Eds.), *Exploring Uncertainty* (pp. 245–269). Springer Fachmedien Wiesbaden. http://doi.org/10.1007/978-3-658-00897-0_10
- Wildemann, H. (2000). *Komplexitätsmanagement: Vertrieb, Produkte, Beschaffung, F & E, Produktion und Administration*. Verlag TCW Transfer-Centrum.
- Williams, T. M. (1999). The need for new paradigms for complex projects. *International Journal of Project Management*, 17(5), 269–273. [http://doi.org/10.1016/S0263-7863\(98\)00047-7](http://doi.org/10.1016/S0263-7863(98)00047-7)
- Williford, J., & Chang, A. (1999). Modeling the FedEx IT division: a system dynamics approach to strategic IT planning. *Journal of Systems and Software*, 46(2), 203–211. [http://doi.org/10.1016/S0164-1212\(99\)00013-8](http://doi.org/10.1016/S0164-1212(99)00013-8)
- Winter, M., Smith, Charles, Morris, Peter, & Cicmil, Svetlana. (2006). Directions for future research in project management: The main findings of a UK government-funded research network. *International Journal of Project Management*, 24(8), 638–649. <http://doi.org/10.1016/j.ijproman.2006.08.009>
- Wood, K. L., Antonsson, E. K., & Beck, J. L. (1990). Representing imprecision in engineering design: Comparing fuzzy and probability calculus. *Research in Engineering Design*, 1(3-4), 187–203. <http://doi.org/10.1007/BF01581211>
- Wynn, D. C., Eckert, C. M., & Clarkson, J. P. (2003). Planning business processes in product development organisations. In *Workshop on Requirements Engineering for Business Process Support (REBPS'03)*. Klagenfurt/Velden, Austria.
- Wynn, D. C., Eckert, C. M., & Clarkson, J. P. (2007). Modelling iteration in engineering design. In *International Conference on Engineering Design, ICED07*. Paris, France.
- Wynn, D. C., Kreimeyer, M., Eben, K., Maurer, M. S., Lindemann, U., & Clarkson, J. P. (2010). Managing complexity by modelling dependencies : proceedings of the 12th International DSM Conference, Cambridge, UK, 22-23 July 2010. In *12th International Dependency and Structure Modelling Conference*. Cambridge, UK: Hanser.
- Xia, W., & Lee, G. (2004). Grasping the complexity of IS development projects. *Communications of the ACM*, 47(5), 68–74. <http://doi.org/10.1145/986213.986215>
- Xia, W., & Lee, G. (2005). Complexity of Information Systems Development Projects: Conceptualization and Measurement Development. *Journal of Management Information Systems*, 22(1), 45–83.
- Yassine, A. A., & Braha, D. (2003). Complex Concurrent Engineering and the Design Structure Matrix Method. *Concurrent Engineering*, 11(3), 165–176. <http://doi.org/10.1177/106329303034503>
- Yassine, A. A., Joglekar, N., Braha, D., Eppinger, S. D., & Whitney, D. (2003). Information

- hiding in product development: The design churn effect. *Research in Engineering Design*, 14(3), 145–161. <http://doi.org/10.1007/s00163-003-0036-2>
- Yassine, A. A., Whitney, D., & Zambito, T. (2001). Assessment of Rework Probabilities for Simulating Product Development Processes using the Design Structure Matrix (DSM). In *Proceedings of DETC '01 ASME 2001 International Design Engineering Technical Conferences Computers and Information in Engineering Conference* (pp. 1–9). Pittsburgh, USA.
- Yin, R. K. (1989). *Case Study Research: Design And Methods* (Applied Social Research Methods) Author: Robert K. Yin, Publisher: Sage Publicat.
- Yin, X. (2009). *Multiscale design under uncertainty*. Northwestern University.

9. Appendix

9.1 Identified literature of literature search

Table 9-1 Identified literature of literature search – part 1

Authors	Title	Source	Volume	Issue	From page	To page	DOI	Year
Lu, SOY; Cai, J	A collaborative design process model in the sociotechnical engineering design framework	AI EDAM-ARTIFICIAL INTELLIGENCE FOR ENGINEERING DESIGN ANALYSIS AND MANUFACTURING	15	1	3	20	10.1017/S089006040115105X	2001
Yassine, AA; Chelst, KR; Falkenburg, DR	A decision analytic framework for evaluating concurrent engineering	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	46	2	144	157	10.1109/17.759142	1999
Cronemyr, P; Ronnback, AC; Eppinger, SD	A decision support tool for predicting the impact of development process improvements	JOURNAL OF ENGINEERING DESIGN	12	3	177	199		2001
Oliver, D.W.	A draft integration of information models: Complement model and Oliver model	Proceedings 1994 Tutorial and Workshop on Systems Engineering of Computer-Based Systems (Cat. No. 94TH0655-1)			44	69	10.1109/ECBS.1994.331684	1994
Tingui Chen; Renbin Xiao	A Dynamic Intelligent Decision Approach to Dependency Modeling of Project Tasks in Complex Engineering System Optimization	Mathematical Problems in Engineering				398123 (12 pp.)	10.1155/2013/398123	2013
Ding-Bang Luh; Yao-Tsung Ko; Chia-Hsiang Ma	A dynamic planning approach for new product development	Concurrent Engineering: Research and Applications	17	1	43	59	10.1177/1063293X09102249	2009
Van Wie, M; Bryant, CR; Bohm, MR; Mcadams, DA; Stone, RB	A model of function-based representations	AI EDAM-ARTIFICIAL INTELLIGENCE FOR ENGINEERING DESIGN ANALYSIS AND MANUFACTURING	19	2	89	111	10.1017/S0890060405050092	2005
Smith, RP; Eppinger, SD	A predictive model of sequential iteration in engineering design	MANAGEMENT SCIENCE	43	8	1104	1120	10.1287/mnsc.43.8.1104	1997
Chen, SJ; Lin, L	A project task coordination model for team organization in concurrent engineering	CONCURRENT ENGINEERING-RESEARCH AND APPLICATIONS	10	3	187	202	10.1106/106329302027637	2002
Soo-Haeng Cho; Eppinger, S.D.	A simulation-based process model for managing complex design projects	IEEE Transactions on Engineering Management	52	3	316	28	10.1109/TEM.2005.850722	2005
Padilla, J.J.; Logan, B.; Sousa-Poza, A.; Keating, C.B.	A system of systems engineering environment to deal with complex situations	2008 IEEE International Conference on System of Systems Engineering (SoSE)			5 pp.	5 pp.	10.1109/SYSOSE.2008.4724208	2008
Levardy, Viktor; Browning, Tyson R.	An Adaptive Process Model to Support Product Development Project Management	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	56	4	600	620	10.1109/TEM.2009.2033144	2009
Hanpeng Zhang; Wanhua Qiu; Hanfei Zhang	An approach to measuring coupled tasks strength and sequencing of coupled tasks in new product development	Concurrent Engineering: Research and Applications	14	4	305	11	10.1177/1063293X06072483	2006
Bashir, Hamdi A.; AlZabdeh, Khalid; Abdo, Jamil	An Eigenvalue Based Approach for Assessing the Decomposability of Interdependent Design Project Tasks	CONCURRENT ENGINEERING-RESEARCH AND APPLICATIONS	17	1	35	42	10.1177/1063293X09102248	2009

Table 9-2 Identified literature of literature search – part 2

Authors	Title	Source	Volume	Issue	From page	To page	DOI	Year
Chen, Shi-Jie (Gary)	An integrated methodological framework for project task coordination and team organization in concurrent engineering	CONCURRENT ENGINEERING-RESEARCH AND APPLICATIONS	13	3	185	197	10.1177/1063293X05056462	2005
Yang, Qing; Yao, Tao; Lu, Ting; Zhang, Bo	An Overlapping-Based Design Structure Matrix for Measuring Interaction Strength and Clustering Analysis	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	61	1	159	170	10.1109/TEM.2013.2267779	2014
Qing Yang; Xiaofeng Zhang; Tao Yao	An overlapping-based process model for managing schedule and cost risk in product development	Concurrent Engineering: Research and Applications	20	1	3	17	10.1177/1063293X12438701	2012
Collins, Shawn T.; Bradley, Joe A.; Yassine, Ali A.	Analyzing Product Development Task Networks to Examine Organizational Change	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	57	3	513	525	10.1109/TEM.2009.2033047	2010
Browning, TR	Applying the design structure matrix to system decomposition and integration problems: A review and new directions	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	48	3	292	306	10.1109/17.946528	2001
GuoRong Chai; ZhiKun Du; Guoxing Zhang; Yanan Su	Calculation of the Cycle and Working Hours for Complex Projects with Iterations	Proceedings of the 2011 International Conference on Information Technology, Computer Engineering and Management Sciences (ICM 2011)			309	12	10.1109/ICM.2011.15	2011
Kannenglesser, Udo	CAN WE ENGINEER BETTER PROCESS MODELS?	ICED 09 - THE 17TH INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, VOL 1: DESIGN PROCESSES			527	538		2009
Yassine, A; Braha, D	Complex concurrent engineering and the design structure matrix method	CONCURRENT ENGINEERING-RESEARCH AND APPLICATIONS	11	3	165	176	10.1177/106329303034503	2003
Helms, Bergen; Shea, Kristina	Computational Synthesis of Product Architectures Based on Object-Oriented Graph Grammars	JOURNAL OF MECHANICAL DESIGN	134	2			10.1115/1.4005592	2012
Hong-Sen Yan; Bin Wang; Duo Xu; Zheng	Computing completion time and optimal scheduling of design activities in concurrent product development	IEEE Transactions on Systems, Man and Cybernetics, Part A (Systems and Humans)	40	1	76	89	10.1109/TSMCA.2009.2030417	2010
Hong Qin Wei	Concurrent design process analysis and optimization for aluminum profile extrusion product development	International Journal of Advanced Manufacturing Technology	33		652	61	10.1007/s00170-007-1025-9	2007
Yassine, A; Whitney, D; Daleiden, S; Lavine, J	Connectivity maps: modeling and analysing relationships in product development processes	JOURNAL OF ENGINEERING DESIGN	14	3	377	394	10.1080/0954482031000091103	2003
Lee, SG; Ong, KL; Khoo, LP	Control and monitoring of concurrent design tasks in a dynamic environment	CONCURRENT ENGINEERING-RESEARCH AND APPLICATIONS	12	1	59	66	10.1177/1063293X04041941	2004
Wang, Zheng; Magee, Christopher L.	Convergence Analysis and Iteration Estimation for a Coupled Design Process With Overlap in Redesign	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	59	4	621	633	10.1109/TEM.2011.2170691	2012

Table 9-3 Identified literature of literature search – part 3

Authors	Title	Source	Volume	Issue	From page	To page	DOI	Year
Kasperek, D.; Maurer, M.	Coupling structural complexity management and system dynamics to represent the dynamic behavior of product development processes	Proceedings of the 2013 7th Annual IEEE Systems Conference (SysCon)			414	419	10.1109/SysCon.2013.6549915	2013
Chen, SJ; Li, L	Decomposition of interdependent task group for concurrent engineering	COMPUTERS & INDUSTRIAL ENGINEERING	44	3	435	459	10.1016/S0360-8352(02)00230-9	2003
Meier, Christoph; Yassine, Ali A.; Browning, Tyson R.	Design process sequencing with competent genetic algorithms	JOURNAL OF MECHANICAL DESIGN	129	6	566	585	10.1115/1.2717224	2007
Pape, Carla; Whitney, Daniel; Henriques, Elsa; Farndon, Rob; Moss, Michael	DEVELOPMENT OF A FRAMEWORK FOR IMPROVING ENGINEERING PROCESSES	PROCEEDINGS OF THE 18TH INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN (ICED 11): IMPACTING SOCIETY THROUGH ENGINEERING DESIGN	1		417	428		2011
Ford, DN; Sterman, JD	Dynamic modeling of product development processes	SYSTEM DYNAMICS REVIEW	14	1	31	68	10.1002/(SICI)1099-1727(199821)14:1<31::AID-SDR141>3.0.CO;2-5	1998
El-Haddad, H. G.; Backar, S. H.; El-Kadeem, R. A.; El-Dardiry, M. A.	Dynamic View of Product Development Process	2012 FIRST INTERNATIONAL CONFERENCE ON INNOVATIVE ENGINEERING SYSTEMS (ICIES)			213	218		2012
Whitfield, RJ; Duffy, AHB; Coates, G; Hills, W	Efficient process optimization	CONCURRENT ENGINEERING-RESEARCH AND APPLICATIONS	11	2	83	92	10.1177/1063293030302082	2003
Enzhen Huang; Shi-Jie Chen	Estimation of project completion time and factors analysis for concurrent engineering project	Concurrent Engineering: Research and Applications	14	4	329	41	10.1177/1063293X06072482	2006
Mittermayer, Herwig; Rodriguez-Monroy, Carlos	Evaluating alternative industrial network organizations and information systems	INDUSTRIAL MANAGEMENT & DATA SYSTEMS	113		77	95	10.1108/02635571311289674	2013
Kang, Changmuk; Hong, Yoo S.	Evaluation of Acceleration Effect of Dynamic Sequencing of Design Process in a Multiproject	JOURNAL OF MECHANICAL DESIGN	131	2			10.1115/1.3066599	2009
Karniel, Arie; Reich, Yoram	Formalizing a Workflow-Net Implementation of Design-Structure-Matrix-Based Process Planning for New Product Development	IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS PART A-SYSTEMS AND HUMANS	41	3	476	491	10.1109/TSMCA.2010.2091954	2011
Karniel, Arie; Reich, Yoram	From DSM-Based Planning to Design Process Simulation: A Review of Process Scheme Logic Verification Issues	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	56	4	636	649	10.1109/TEM.2009.2032032	2009
Grabowski, H.; Lossack, R.; Kunze, H.	Functional modeling: Representation of dynamic aspects in function structures	Proceedings of the Twelfth International Florida AI Research Society Conference			399	404		1999
Smith, RP; Eppinger, SD	Identifying controlling features of engineering design iteration	MANAGEMENT SCIENCE	43	3	276	293	10.1287/mnsc.43.3.276	1997

Table 9-4 Identified literature of literature search – part 4

Authors	Title	Source	Volume	Issue	From page	To page	DOI	Year
Leon, Hilda C. Martinez; Farris, Jennifer A.; Letens, Geert	Improving Product Development Performance Through Iteration Front-Loading	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	60	3	552	565	10.1109/TEM.2012.2228205	2013
Chen, CH; Khoo, LP; Jiao, L	Information deduction approach through quality function deployment for the quantification of the dependency	INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH	42	21	4623	4637	10.1080/00207540410001721727	2004
Maier, MW	Integrated modeling: A unified approach to system engineering	JOURNAL OF SYSTEMS AND SOFTWARE	32	2	101	119	10.1016/0164-1212(94)00063-8	1996
Peikovic, I.; Peikovic, V.	Interval Matrix Models of Design Iteration	2010 International Conference on Electronics and Information Engineering (ICEIE 2010)			20	4	10.1109/ICEIE.2010.5559830	2010
Browning, TR; Eppinger, SD	Modeling impacts of process architecture on cost and schedule risk in product development	IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT	49	4	428	442	10.1109/TEM.2002.806709	2002
Li, Ying; Taylor, Timothy R. B.	Modeling the Impact of Design Rework on Transportation Infrastructure Construction Project Performance	JOURNAL OF CONSTRUCTION ENGINEERING AND MANAGEMENT	140	9			10.1061/(ASCE)CO.1943-7862.0000876	2014
Ng, T.S.A.	Modelling large-scale manpower dynamics: An expert systems approach	Proceedings of the 2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS)			159	64	10.1109/ICIS.2011.6070320	2011
Kim, Dohyeon	On representations and dynamic analysis of concurrent engineering design	JOURNAL OF ENGINEERING DESIGN	18	3	265	277	10.1080/09544820600822097	2007
Suss, Samuel; Thomson, Vince	Optimal design processes under uncertainty and reciprocal dependency	JOURNAL OF ENGINEERING DESIGN	23		826	848	10.1080/09544828.2012.704546	2012
Hossain, M. A.; Chua, D. K. H.; Liu, Z.	Optimizing Concurrent Execution of Design Activities with Minimum Redesign	JOURNAL OF COMPUTING IN CIVIL ENGINEERING	26	3	409	420	10.1061/(ASCE)CP.1943-5487.0000150	2012
Joglekar, NR; Ford, DN	Product development resource allocation with foresight	EUROPEAN JOURNAL OF OPERATIONAL RESEARCH	160	1	72	87	10.1016/j.ejor.2003.06.021	2005
Feng Tao; Chen Bing; Zhang Dinghua; Li Shan	Products Development Process Based on DSM and Its Optimization	2009 INTERNATIONAL CONFERENCE ON MEASURING TECHNOLOGY AND MECHATRONICS AUTOMATION, VOL III			689	692	10.1109/ICMTMA.2009.567	2009
D'Avino, G; Dondo, P.; lo Storto, C; Zezza, V	Reducing ambiguity and uncertainty during new product development: A system dynamics based approach	Technology Management: A Unifying Discipline for Merging the Boundaries			538	549		2005
Xiao, Renbin; Chen, Tinggui; Ju, Chunhua	RESEARCH ON PRODUCT DEVELOPMENT ITERATIONS BASED ON FEEDBACK CONTROL THEORY IN A DYNAMIC ENVIRONMENT	INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING INFORMATION AND CONTROL	7	5B	2669	2688		2011

Table 9-5 Identified literature of literature search – part 5

Authors	Title	Source	Volume	Issue	From page	To page	DOI	Year
Karniel, Arie; Reich, Yoram	Simulating design processes with self-iteration activities based on DSM planning	2007 International Conference on Systems Engineering and Modeling, Proceedings			33	41	10.1109/ICSEM.2007.373331	2007
Maier, Jakob F.; Wynn, David C.; Biedermann, Wieland; Lindemann, Levitt, RE; Thomsen, J;	Simulating progressive iteration, rework and change propagation to prioritise design tasks	RESEARCH IN ENGINEERING DESIGN	25	4	283	307	10.1007/s00163-014-0174-8	2014
Christiansen, TR; Kunz, JC; Jin, Y; Niass, C	Simulating project work processes and organizations: Toward a micro-contingency theory of organizational design	MANAGEMENT SCIENCE	45	11	1479	1495	10.1287/mnsc.45.11.1479	1999
Kasperek, Daniel; Malsenbacher, Sebastian; Maurer, Maik	Structure-based Analysis of Dynamic Engineering Process Behavior	2014 8TH ANNUAL IEEE SYSTEMS CONFERENCE (SYSCON)			83	88		2014
Ahmadi, R; Roemer, TA; Wang, RH	Structuring product development processes	EUROPEAN JOURNAL OF OPERATIONAL RESEARCH	130	3	539	558	10.1016/S0377-2217(99)00412-9	2001
Kara, Sami; Kays, Berman	The effect of the learning process in concurrent engineering projects	CONCURRENT ENGINEERING-RESEARCH AND APPLICATIONS	13	3	209	217	10.1177/1063293X05056473	2005
Agyapong-Kodua, K.; Marzano, A.; Ratchev, S.	The Integrated Use of Enterprise and System Dynamics Modelling Techniques in Manufacturing Enterprises	45TH CIRP CONFERENCE ON MANUFACTURING SYSTEMS 2012	3		543	548	10.1016/j.procir.2012.07.093	2012
Keller, R.; Flanagan, T.; Eckert, C.M.; Clarkson, P.J.	Two sides of the story: visualising products and processes in engineering design	Information Visualization			6 pp.	6 pp.		2006
Zhang Wei-min; Zhou Bo-sheng; Luo Wen-je	Extended analytic model of sequential iterative processes	Computer Engineering	34	6	273	5		2008

9.2 Selection of relevant challenges from Sterman (2000)

An extensive collection of information on System Dynamics can be found in Sterman (2000). It comprises a large set of heuristics and models which can be taken as a basis for quantification. In particular Sterman explains 79 challenges of System Dynamics and provides potential solutions in form of heuristics and examples. The table illustrates 66 of these challenges which may be most relevant for the approach presented in this thesis. As the solutions are described in detail in (Sterman, 2000), this thesis refers to original reference for detailed information on the particular heuristics.

Table 9-6 Selection of relevant challenges from (Sterman, 2000)

Dynamics of Multiple-Loop Systems	Finding Formulation Flaws
Identifying Feedback Structure from System Behavior	Multiple Nonlinear Effects
Assigning Link Polarities	Floating Goals
Identifying Link and Loop Polarity	Goal Formation with Internal and External Inputs
Employee Motivation	Finding the Optimal Mix of Capital and Labor
Process Improvement	Preventing Negative Stocks
Policy Analysis with Causal Diagrams	Formulating Nonlinear Functions
Speculative Bubbles	Refining Table Functions with Qualitative Data
Identifying the Feedback Structure of Policy Resistance	Critiquing Nonlinear Functions
Identifying Stocks and Flows	Formulating the Error Rate
Adding Stock and Flow Structure to Causal Diagrams	Testing the Full Model
Linking Stock and Flow Structure with Feedback	Exploring Noise
Modifying Stock and Flow Maps	Policy Design in the Market Growth Model
Disaggregation	Extrapolation and Stability
Graphical Integration	Exploring Amplification
Graphical Differentiation	Exploring the Stock Management Structure
Goal-Seeking Behavior	Simultaneous Initial Conditions
Nonlinear Birth and Death Rates	Reengineering the Supply Chain
Exploring the SIR Model	Mental Simulation of Inventory Management with Labor
The Efficacy of Immunization Programs	Explaining Oscillations
Extending the SIR Model	Policy Design to Enhance Stability
Modeling Design Wins in the Semiconductor Industry	Reengineering a Manufacturing Firm for Enhanced Stability
The Bass Diffusion Model	The Costs of Instability
Modeling the Life Cycle of Durable Products	Adding Training and Experience
Identifying Path Dependence	Intended Rationality of the Investment Process
Formulating a Dynamic Hypothesis for the VCR Industry	Sensitivity to Uncertainty in Parameters
Policy Analysis	Sensitivity to Structural Changes
Extending the Model	Implementing Structural Changes-Modeling Livestock Markets
Duration and Dynamics of Delays	Policy Analysis
The Dynamics of Experience and Learning	Extreme Condition Tests
Response of Delays to Changing Delay Times	Model Testing
The Interactions of Training Delays and Growth	Putting System Dynamics Into Action
Coflows	Choosing a Time Step

9.3 Supervised student theses

The following theses were supervised and guided by the author and to some extent became content of this thesis:

- Produktentwicklung. (2013). Semesterarbeit Lichtenberg: Strukturbasierte Modellierung und Bewertung eines Product-Service-Systems in der frühen Phase (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014a). Bachelorarbeit Bermond: Strukturbasierte System Dynamics-Modellierung zur Prozessoptimierung und -validierung (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014b). Masterarbeit Lindinger: Entwicklung eines Leitfadens zur strukturbasierten System Dynamics Simulation (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014c). Semester Thesis Schmidt: A guideline for adapted System Dynamics modeling of rework cycles in engineering design processes (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014d). Semesterarbeit Lindinger: Multiple Domain Matrix basierte Implementierung von Entwicklungsprozessen in System Dynamics (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014e). Semesterarbeit Berger: Analyse des MAN Benchmarking-Prozesses mittels strukturbasierter System Dynamics Modellierung (not published, confidential). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2014f). Semesterarbeit Lichtenberg: Dynamische Verhaltenssimulation mittels System Dynamics (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2015a). Bachelor Thesis Steinkirchner: A Structured Approach for Identifying Relevant Literature in Engineering Sciences (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2015b). Bachelorarbeit Bierig: Kombination aus “Strukturellem Komplexitätsmanagement” und “System Dynamics” - Gestaltung der Inbetriebnahme des Großprojektes CALA (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2015c). Semester Thesis Schenk: Dynamic Structure Simulation - Strukturbasierte Simulation eines Entwicklungsprozesses der frühen Phase komplexer Großprojekte am Beispiel der Nutzfahrzeugindustrie (not published, confidential). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.
- Produktentwicklung. (2015d). Semesterarbeit Großmann: Methode zum verbesserten Komplexitätsverständnis in strukturellen und dynamischen Modellen (not published). Munich, Germany: Lehrstuhl für Produktentwicklung; Technische Universität München.

10. List of dissertations

Institute of Product Development

Technical University of Munich, Boltzmannstraße 15, 85748 Garching

Dissertations supervised by

- Prof. Dr.-Ing. W. Rodenacker,
- Prof. Dr.-Ing. K. Ehrlenspiel and
- Prof. Dr.-Ing. U. Lindemann

- D1 COLLIN, H.:
Entwicklung eines Einwalzenkalenders nach einer systematischen Konstruktionsmethode. München: TU, Diss. 1969.
- D2 OTT, J.:
Untersuchungen und Vorrichtungen zum Offen-End-Spinnen.
München: TU, Diss. 1971.
- D3 STEINWACHS, H.:
Informationsgewinnung an bandförmigen Produkten für die Konstruktion der Produktmaschine.
München: TU, Diss. 1971.
- D4 SCHMETTOW, D.:
Entwicklung eines Rehabilitationsgerätes für Schwerstkörperbehinderte.
München: TU, Diss. 1972.
- D5 LUBITZSCH, W.:
Die Entwicklung eines Maschinensystems zur Verarbeitung von chemischen Endlosfasern.
München: TU, Diss. 1974.
- D6 SCHEITENBERGER, H.:
Entwurf und Optimierung eines Getriebesystems für einen Rotationsquerschneider mit allgemeingültigen Methoden.
München: TU, Diss. 1974.
- D7 BAUMGARTH, R.:
Die Vereinfachung von Geräten zur Konstanthaltung physikalischer Größen.
München: TU, Diss. 1976.
- D8 MAUDERER, E.:
Beitrag zum konstruktionsmethodischen Vorgehen durchgeführt am Beispiel eines Hochleistungsschalter-Antriebs.
München: TU, Diss. 1976.
- D9 SCHÄFER, J.:
Die Anwendung des methodischen Konstruierens auf verfahrenstechnische Aufgabenstellungen.
München: TU, Diss. 1977.
- D10 WEBER, J.:
Extruder mit Feststoffpumpe – Ein Beitrag zum Methodischen Konstruieren.
München: TU, Diss. 1978.
- D11 HEISIG, R.:
Längencodierer mit Hilfsbewegung.
München: TU, Diss. 1979.
- D12 KIEWERT, A.:
Systematische Erarbeitung von Hilfsmitteln zum kostenarmen Konstruieren.
München: TU, Diss. 1979.

- D13 LINDEMANN, U.:
Systemtechnische Betrachtung des Konstruktionsprozesses unter besonderer Berücksichtigung der Herstellkostenbeeinflussung beim Festlegen der Gestalt.
Düsseldorf: VDI-Verlag 1980. (Fortschritt-Berichte der VDI-Zeitschriften Reihe 1, Nr. 60).
Zugl. München: TU, Diss. 1980.
- D14 NJOYA, G.:
Untersuchungen zur Kinematik im Wälzlager bei synchron umlaufenden Innen- und Außenringen.
Hannover: Universität, Diss. 1980.
- D15 HENKEL, G.:
Theoretische und experimentelle Untersuchungen ebener konzentrisch gewellter Kreisringmembranen.
Hannover: Universität, Diss. 1980.
- D16 BALKEN, J.:
Systematische Entwicklung von Gleichlaufgelenken.
München: TU, Diss. 1981.
- D17 PETRA, H.:
Systematik, Erweiterung und Einschränkung von Lastausgleichslösungen für Standgetriebe mit zwei Leistungswegen – Ein Beitrag zum methodischen Konstruieren.
München: TU, Diss. 1981.
- D18 BAUMANN, G.:
Ein Kosteninformationssystem für die Gestaltungsphase im Betriebsmittelbau.
München: TU, Diss. 1982.
- D19 FISCHER, D.:
Kostenanalyse von Stirnzahnrädern. Erarbeitung und Vergleich von Hilfsmitteln zur Kostenfrüherkennung.
München: TU, Diss. 1983.
- D20 AUGUSTIN, W.:
Sicherheitstechnik und Konstruktionsmethodiken – Sicherheitsgerechtes Konstruieren.
Dortmund: Bundesanstalt für Arbeitsschutz 1985. Zugl. München: TU, Diss. 1984.
- D21 RUTZ, A.:
Konstruieren als gedanklicher Prozess.
München: TU, Diss. 1985.
- D22 SAUERMANN, H. J.:
Eine Produktkostenplanung für Unternehmen des Maschinenbaues.
München: TU, Diss. 1986.
- D23 HAFNER, J.:
Entscheidungshilfen für das kostengünstige Konstruieren von Schweiß- und Gussgehäusen.
München: TU, Diss. 1987.
- D24 JOHN, T.:
Systematische Entwicklung von homokinetischen Wellenkupplungen.
München: TU, Diss. 1987.
- D25 FIGEL, K.:
Optimieren beim Konstruieren.
München: Hanser 1988. Zugl. München: TU, Diss. 1988 u. d. T.: Figel, K.: Integration automatisierter Optimierungsverfahren in den rechnerunterstützten Konstruktionsprozess.

- D26 TROPSCHUH, P. F.:
Rechnerunterstützung für das Projektieren mit Hilfe eines wissensbasierten Systems.
München: Hanser 1989. (Konstruktionstechnik München, Band 1). Zugl. München: TU, Diss. 1988 u. d. T.: Tropschuh, P. F.: Rechnerunterstützung für das Projektieren am Beispiel Schiffsgetriebe.
- D27 PICKEL, H.:
Kostenmodelle als Hilfsmittel zum Kostengünstigen Konstruieren.
München: Hanser 1989. (Konstruktionstechnik München, Band 2). Zugl. München: TU, Diss. 1988.
- D28 KITSTEINER, H.-J.:
Die Auswahl und Gestaltung von kostengünstigen Welle-Nabe-Verbindungen.
München: Hanser 1990. (Konstruktionstechnik München, Band 3). Zugl. München: TU, Diss. 1989.
- D29 HILLEBRAND, A.:
Ein Kosteninformationssystem für die Neukonstruktion mit der Möglichkeit zum Anschluss an ein CAD-System.
München: Hanser 1991. (Konstruktionstechnik München, Band 4). Zugl. München: TU, Diss. 1990.
- D30 DYLLA, N.:
Denk- und Handlungsabläufe beim Konstruieren.
München: Hanser 1991. (Konstruktionstechnik München, Band 5). Zugl. München: TU, Diss. 1990.
- D31 MÜLLER, R.:
Datenbankgestützte Teileverwaltung und Wiederholteilsuche.
München: Hanser 1991. (Konstruktionstechnik München, Band 6). Zugl. München: TU, Diss. 1990.
- D32 NEESE, J.:
Methodik einer wissensbasierten Schadenanalyse am Beispiel Wälzlagerungen.
München: Hanser 1991. (Konstruktionstechnik München, Band 7). Zugl. München: TU, Diss. 1991.
- D33 SCHAAL, S.:
Integrierte Wissensverarbeitung mit CAD – Am Beispiel der konstruktionsbegleitenden Kalkulation.
München: Hanser 1992. (Konstruktionstechnik München, Band 8). Zugl. München: TU, Diss. 1991.
- D34 BRAUNSPERGER, M.:
Qualitätssicherung im Entwicklungsablauf – Konzept einer präventiven Qualitätssicherung für die Automobilindustrie.
München: Hanser 1993. (Konstruktionstechnik München, Band 9). Zugl. München: TU, Diss. 1992.
- D35 FEICHTER, E.:
Systematischer Entwicklungsprozess am Beispiel von elastischen Radialversatzkupplungen.
München: Hanser 1994. (Konstruktionstechnik München, Band 10). Zugl. München: TU, Diss. 1992.
- D36 WEINBRENNER, V.:
Produktlogik als Hilfsmittel zum Automatisieren von Varianten- und Anpassungskonstruktionen.
München: Hanser 1994. (Konstruktionstechnik München, Band 11). Zugl. München: TU, Diss. 1993.
- D37 WACH, J. J.:
Problemspezifische Hilfsmittel für die Integrierte Produktentwicklung.
München: Hanser 1994. (Konstruktionstechnik München, Band 12). Zugl. München: TU, Diss. 1993.
- D38 LENK, E.:
Zur Problematik der technischen Bewertung.
München: Hanser 1994. (Konstruktionstechnik München, Band 13). Zugl. München: TU, Diss. 1993.
- D39 STUFFER, R.:
Planung und Steuerung der Integrierten Produktentwicklung.
München: Hanser 1994. (Konstruktionstechnik München, Band 14). Zugl. München: TU, Diss. 1993.
- D40 SCHIEBELER, R.:
Kostengünstig Konstruieren mit einer rechnergestützten Konstruktionsberatung.
München: Hanser 1994. (Konstruktionstechnik München, Band 15). Zugl. München: TU, Diss. 1993.

- D41 BRUCKNER, J.:
Kostengünstige Wärmebehandlung durch Entscheidungsunterstützung in Konstruktion und Härterei.
München: Hanser 1994. (Konstruktionstechnik München, Band 16). Zugl. München: TU, Diss. 1993.
- D42 WELLNIAK, R.:
Das Produktmodell im rechnerintegrierten Konstruktionsarbeitsplatz.
München: Hanser 1994. (Konstruktionstechnik München, Band 17). Zugl. München: TU, Diss. 1994.
- D43 SCHLÜTER, A.:
Gestaltung von Schnappverbindungen für montagegerechte Produkte.
München: Hanser 1994. (Konstruktionstechnik München, Band 18). Zugl. München: TU, Diss. 1994.
- D44 WOLFRAM, M.:
Feature-basiertes Konstruieren und Kalkulieren.
München: Hanser 1994. (Konstruktionstechnik München, Band 19). Zugl. München: TU, Diss. 1994.
- D45 STOLZ, P.:
Aufbau technischer Informationssysteme in Konstruktion und Entwicklung am Beispiel eines elektronischen Zeichnungsarchives.
München: Hanser 1994. (Konstruktionstechnik München, Band 20). Zugl. München: TU, Diss. 1994.
- D46 STOLL, G.:
Montagegerechte Produkte mit feature-basiertem CAD.
München: Hanser 1994. (Konstruktionstechnik München, Band 21). Zugl. München: TU, Diss. 1994.
- D47 STEINER, J. M.:
Rechnergestütztes Kostensenken im praktischen Einsatz.
Aachen: Shaker 1996. (Konstruktionstechnik München, Band 22). Zugl. München: TU, Diss. 1995.
- D48 HUBER, T.:
Senken von Montagezeiten und -kosten im Getriebebau.
München: Hanser 1995. (Konstruktionstechnik München, Band 23). Zugl. München: TU, Diss. 1995.
- D49 DANNER, S.:
Ganzheitliches Anforderungsmanagement für marktorientierte Entwicklungsprozesse.
Aachen: Shaker 1996. (Konstruktionstechnik München, Band 24). Zugl. München: TU, Diss. 1996.
- D50 MERAT, P.:
Rechnergestützte Auftragsabwicklung an einem Praxisbeispiel.
Aachen: Shaker 1996. (Konstruktionstechnik München, Band 25). Zugl. München: TU, Diss. 1996 u. d. T.: MERAT, P.: Rechnergestütztes Produktleitsystem
- D51 AMBROSY, S.:
Methoden und Werkzeuge für die integrierte Produktentwicklung.
Aachen: Shaker 1997. (Konstruktionstechnik München, Band 26). Zugl. München: TU, Diss. 1996.
- D52 GIAPOULIS, A.:
Modelle für effektive Konstruktionsprozesse.
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 27). Zugl. München: TU, Diss. 1996.
- D53 STEINMEIER, E.:
Realisierung eines systemtechnischen Produktmodells – Einsatz in der Pkw-Entwicklung
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 28). Zugl. München: TU, Diss. 1998.
- D54 KLEEDÖRFER, R.:
Prozess- und Änderungsmanagement der Integrierten Produktentwicklung.
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 29). Zugl. München: TU, Diss. 1998.
- D55 GÜNTHER, J.:
Individuelle Einflüsse auf den Konstruktionsprozess.
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 30). Zugl. München: TU, Diss. 1998.

- D56 BIERSACK, H.:
Methode für Kraftereinleitungsstellenkonstruktion in Blechstrukturen.
München: TU, Diss. 1998.
- D57 IRLINGER, R.:
Methoden und Werkzeuge zur nachvollziehbaren Dokumentation in der Produktentwicklung.
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 31). Zugl. München: TU, Diss. 1999.
- D58 EILETZ, R.:
Zielkonfliktmanagement bei der Entwicklung komplexer Produkte – am Bsp. PKW-Entwicklung.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 32). Zugl. München: TU, Diss. 1999.
- D59 STÖSSER, R.:
Zielkostenmanagement in integrierten Produkterstellungsprozessen.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 33). Zugl. München: TU, Diss. 1999.
- D60 PHLEPS, U.:
Recyclinggerechte Produktdefinition – Methodische Unterstützung für Upgrading und Verwertung.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 34). Zugl. München: TU, Diss. 1999.
- D61 BERNARD, R.:
Early Evaluation of Product Properties within the Integrated Product Development.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 35). Zugl. München: TU, Diss. 1999.
- D62 ZANKER, W.:
Situative Anpassung und Neukombination von Entwicklungsmethoden.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 36). Zugl. München: TU, Diss. 1999.

Reihe Produktentwicklung München

- D63 ALLMANSBERGER, G.:
Erweiterung der Konstruktionsmethodik zur Unterstützung von Änderungsprozessen in der Produktentwicklung.
München: Dr. Hut 2001. (Produktentwicklung München, Band 37). Zugl. München: TU, Diss. 2000.
- D64 ASSMANN, G.:
Gestaltung von Änderungsprozessen in der Produktentwicklung.
München: Utz 2000. (Produktentwicklung München, Band 38). Zugl. München: TU, Diss. 2000.
- D65 BICHLMAIER, C.:
Methoden zur flexiblen Gestaltung von integrierten Entwicklungsprozessen.
München: Utz 2000. (Produktentwicklung München, Band 39). Zugl. München: TU, Diss. 2000.
- D66 DEMERS, M. T.
Methoden zur dynamischen Planung und Steuerung von Produktentwicklungsprozessen.
München: Dr. Hut 2000. (Produktentwicklung München, Band 40). Zugl. München: TU, Diss. 2000.
- D67 STETTER, R.:
Method Implementation in Integrated Product Development.
München: Dr. Hut 2000. (Produktentwicklung München, Band 41). Zugl. München: TU, Diss. 2000.
- D68 VIERTLBÖCK, M.:
Modell der Methoden- und Hilfsmiteleföhrung im Bereich der Produktentwicklung.
München: Dr. Hut 2000. (Produktentwicklung München, Band 42). Zugl. München: TU, Diss. 2000.
- D69 COLLIN, H.:
Management von Produkt-Informationen in kleinen und mittelständischen Unternehmen.
München: Dr. Hut 2001. (Produktentwicklung München, Band 43). Zugl. München: TU, Diss. 2001.

- D70 REISCHL, C.:
Simulation von Produktkosten in der Entwicklungsphase.
München: Dr. Hut 2001. (Produktentwicklung München, Band 44). Zugl. München: TU, Diss. 2001.
- D71 GAUL, H.-D.:
Verteilte Produktentwicklung - Perspektiven und Modell zur Optimierung.
München: Dr. Hut 2001. (Produktentwicklung München, Band 45). Zugl. München: TU, Diss. 2001.
- D72 GIERHARDT, H.:
Global verteilte Produktentwicklungsprojekte – Ein Vorgehensmodell auf der operativen Ebene.
München: Dr. Hut 2002. (Produktentwicklung München, Band 46). Zugl. München: TU, Diss. 2001.
- D73 SCHOEN, S.:
Gestaltung und Unterstützung von Community of Practice.
München: Utz 2000. (Produktentwicklung München, Band 47). Zugl. München: TU, Diss. 2000.
- D74 BENDER, B.:
Zielorientiertes Kooperationsmanagement.
München: Dr. Hut 2001. (Produktentwicklung München, Band 48). Zugl. München: TU, Diss. 2001.
- D75 SCHWANKL, L.:
Analyse und Dokumentation in den frühen Phasen der Produktentwicklung.
München: Dr. Hut 2002. (Produktentwicklung München, Band 49). Zugl. München: TU, Diss. 2002.
- D76 WULF, J.:
Elementarmethoden zur Lösungssuche.
München: Dr. Hut 2002. (Produktentwicklung München, Band 50). Zugl. München: TU, Diss. 2002.
- D77 MÖRTL, M.:
Entwicklungsmanagement für langlebige, upgradunggerechte Produkte.
München: Dr. Hut 2002. (Produktentwicklung München, Band 51). Zugl. München: TU, Diss. 2002.
- D78 GERST, M.:
Strategische Produktentscheidungen in der integrierten Produktentwicklung.
München: Dr. Hut 2002. (Produktentwicklung München, Band 52). Zugl. München: TU, Diss. 2002.
- D79 AMFT, M.:
Phasenübergreifende bidirektionale Integration von Gestaltung und Berechnung.
München: Dr. Hut 2003. (Produktentwicklung München, Band 53). Zugl. München: TU, Diss. 2002.
- D80 FÖRSTER, M.:
Variantenmanagement nach Fusionen in Unternehmen des Anlagen- und Maschinenbaus.
München: TU, Diss. 2003.
- D81 GRAMANN, J.:
Problemmodelle und Bionik als Methode.
München: Dr. Hut 2004. (Produktentwicklung München, Band 55). Zugl. München: TU, Diss. 2004.
- D82 PULM, U.:
Eine systemtheoretische Betrachtung der Produktentwicklung.
München: Dr. Hut 2004. (Produktentwicklung München, Band 56). Zugl. München: TU, Diss. 2004.
- D83 HUTTERER, P.:
Reflexive Dialoge und Denkbausteine für die methodische Produktentwicklung.
München: Dr. Hut 2005. (Produktentwicklung München, Band 57). Zugl. München: TU, Diss. 2005.
- D84 FUCHS, D.:
Konstruktionsprinzipien für die Problemanalyse in der Produktentwicklung.
München: Dr. Hut 2006. (Produktentwicklung München, Band 58). Zugl. München: TU, Diss. 2005.
- D85 PACHE, M.:
Sketching for Conceptual Design.
München: Dr. Hut 2005. (Produktentwicklung München, Band 59). Zugl. München: TU, Diss. 2005.

- D86 BRAUN, T.:
Methodische Unterstützung der strategischen Produktplanung in einem mittelständisch geprägten Umfeld.
München: Dr. Hut 2005. (Produktentwicklung München, Band 60). Zugl. München: TU, Diss. 2005.
- D87 JUNG, C.:
Anforderungsklä rung in interdisziplinärer Entwicklungsumgebung.
München: Dr. Hut 2006. (Produktentwicklung München, Band 61). Zugl. München: TU, Diss. 2006.
- D88 HEBLING, T.:
Einführung der Integrierten Produktpolitik in kleinen und mittelständischen Unternehmen.
München: Dr. Hut 2006. (Produktentwicklung München, Band 62). Zugl. München: TU, Diss. 2006.
- D89 STRICKER, H.:
Bionik in der Produktentwicklung unter der Berücksichtigung menschlichen Verhaltens.
München: Dr. Hut 2006. (Produktentwicklung München, Band 63). Zugl. München: TU, Diss. 2006.
- D90 NIßL, A.:
Modell zur Integration der Zielkostenverfolgung in den Produktentwicklungsprozess.
München: Dr. Hut 2006. (Produktentwicklung München, Band 64). Zugl. München: TU, Diss. 2006.
- D91 MÜLLER, F.:
Intuitive digitale Geometriemodellierung in frühen Entwicklungsphasen.
München: Dr. Hut 2007. (Produktentwicklung München, Band 65). Zugl. München: TU, Diss. 2006.
- D92 ERDELL, E.:
Methodenanwendung in der Hochbauplanung – Ergebnisse einer Schwachstellenanalyse.
München: Dr. Hut 2006. (Produktentwicklung München, Band 66). Zugl. München: TU, Diss. 2006.
- D93 GAHR, A.:
Pfadkostenrechnung individualisierter Produkte.
München: Dr. Hut 2006. (Produktentwicklung München, Band 67). Zugl. München: TU, Diss. 2006.
- D94 RENNER, I.:
Methodische Unterstützung funktionsorientierter Baukastenentwicklung am Beispiel Automobil.
München: Dr. Hut 2007 (Reihe Produktentwicklung) Zugl. München: TU, Diss. 2007.
- D95 PONN, J.:
Situative Unterstützung der methodischen Konzeptentwicklung technischer Produkte.
München: Dr. Hut 2007 (Reihe Produktentwicklung) Zugl. München: TU, Diss. 2007.
- D96 HERFELD, U.:
Matrix-basierte Verknüpfung von Komponenten und Funktionen zur Integration von Konstruktion und numerischer Simulation.
München: Dr. Hut 2007. (Produktentwicklung München, Band 70). Zugl. München: TU, Diss. 2007.
- D97 SCHNEIDER, S.:
Model for the evaluation of engineering design methods.
München: Dr. Hut 2008 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D98 FELGEN, L.:
Systemorientierte Qualitätssicherung für mechatronische Produkte.
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D99 GRIEB, J.:
Auswahl von Werkzeugen und Methoden für verteilte Produktentwicklungsprozesse.
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D100 MAURER, M.:
Structural Awareness in Complex Product Design.
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.

- D101 BAUMBERGER, C.:
Methoden zur kundenspezifischen Produktdefinition bei individualisierten Produkten.
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D102 KEIJZER, W.:
Wandlungsfähigkeit von Entwicklungsnetzwerken – ein Modell am Beispiel der Automobilindustrie.
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D103 LORENZ, M.:
Handling of Strategic Uncertainties in Integrated Product Development.
München: Dr. Hut 2009 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2008.
- D104 KREIMEYER, M.:
Structural Measurement System for Engineering Design Processes.
München: Dr. Hut 2010 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2009.
- D105 DIEHL, H.:
Systemorientierte Visualisierung disziplinübergreifender Entwicklungsabhängigkeiten mechatronischer Automobilsysteme.
München: Dr. Hut 2009 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2009.
- D106 DICK, B.:
Untersuchung und Modell zur Beschreibung des Einsatzes bildlicher Produktmodelle durch Entwicklerteams in der Lösungssuche.
München: Dr. Hut 2009 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2009.
- D107 GAAG, A.:
Entwicklung einer Ontologie zur funktionsorientierten Lösungssuche in der Produktentwicklung.
München: Dr. Hut 2010 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2010.
- D108 ZIRKLER, S.:
Transdisziplinäres Zielkostenmanagement komplexer mechatronischer Produkte.
München: Dr. Hut 2010 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2010.
- D109 LAUER, W.:
Integrative Dokumenten- und Prozessbeschreibung in dynamischen Produktentwicklungsprozessen.
München: Dr. Hut 2010 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2010.
- D110 MEIWALD, T.:
Konzepte zum Schutz vor Produktpiraterie und unerwünschtem Know-how-Abfluss.
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D111 ROELOFSEN, J.:
Situationsspezifische Planung von Produktentwicklungsprozessen.
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D112 PETERMANN, M.:
Schutz von Technologiewissen in der Investitionsgüterindustrie.
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D113 GORBEA, C.:
Vehicle Architecture and Lifecycle Cost Analysis in a New Age of Architectural Competition.
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D114 FILOUS, M.:
Lizenzierungsgerechte Produktentwicklung – Ein Leitfaden zur Integration lizenzierungsrelevanter Aktivitäten in Produktentstehungsprozessen des Maschinen- und Anlagenbaus.
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D115 ANTON, T.:
Entwicklungs- und Einführungsmethodik für das Projektierungswerkzeug Pneumatiksimulation.
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.

- D116 KESPER, H.:
Gestaltung von Produktvariantenspektren mittels matrixbasierter Methoden.
München: Dr. Hut 2012 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2012.
- D117 KIRSCHNER, R.:
Methodische Offene Produktentwicklung.
München: TU, Diss. 2012.
- D118 HEPPELE, C.:
Planung lebenszyklusgerechter Leistungsbündel.
München: Dr. Hut 2013 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2013.
- D119 HELLENBRAND, D.:
Transdisziplinäre Planung und Synchronisation mechatronischer Produktentwicklungsprozesse.
München: Dr. Hut 2013 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2013.
- D120 EBERL, T.:
Charakterisierung und Gestaltung des Fahr-Erlebens der Längsführung von Elektrofahrzeugen.
München: TU, Diss. 2014.
- D121 KAIN, A.:
Methodik zur Umsetzung der Offenen Produktentwicklung.
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D122 ILIE, D.:
Systematisiertes Ziele- und Anforderungsmanagement in der Fahrzeugentwicklung.
München: Dr. Hut 2013 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2013.
- D123 HELTEN, K.:
Einführung von Lean Development in mittelständische Unternehmen - Beschreibung, Erklärungsansatz
und Handlungsempfehlungen.
München: Dr. Hut 2015 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D124 SCHRÖER, B.:
Lösungskomponente Mensch. Nutzerseitige Handlungsmöglichkeiten als Bausteine für die kreative
Entwicklung von Interaktionslösungen.
München: TU, Diss. 2014.
- D125 KORTLER, S.:
Absicherung von Eigenschaften komplexer und variantenreicher Produkte in der Produktentwicklung.
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D126 KOHN, A.:
Entwicklung einer Wissensbasis für die Arbeit mit Produktmodellen.
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D127 FRANKE, S.:
Strategieorientierte Vorentwicklung komplexer Produkte – Prozesse und Methoden zur zielgerichteten
Komponentenentwicklung am Beispiel Pkw.
Göttingen: Cuvillier, E 2014. Zugl. München: TU, Diss. 2014.
- D128 HOOSHMAND, A.:
Solving Engineering Design Problems through a Combination of Generative Grammars and Simulations.
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D129 KISSEL, M.:
Mustererkennung in komplexen Produktportfolios.
München: TU, Diss. 2014.
- D130 NIES, B.:
Nutzungsgerechte Dimensionierung des elektrischen Antriebssystems für Plug-In Hybride.
München: TU, Diss. 2014.

- D131 KIRNER, K.:
Zusammenhang zwischen Leistung in der Produktentwicklung und Variantenmanagement – Einflussmodell und Analyseverfahren.
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D132 BIEDERMANN, W.:
A minimal set of network metrics for analysing mechatronic product concepts.
München: TU, Diss. 2015.
- D133 SCHENKL, S.:
Wissensorientierte Entwicklung von Produkt-Service-Systemen.
München: TU, Diss. 2015.
- D134 SCHRIEVERHOFF, P.:
Valuation of Adaptability in System Architecture.
München: Dr. Hut 2015 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D135 METZLER, T.:
Models and Methods for the Systematic Integration of Cognitive Functions into Product Concepts.
München: Dr. Hut 2016 (Reihe Produktentwicklung).
- D136 DEUBZER, F.:
A Method for Product Architecture Management in Early Phases of Product Development.
München: TU, Diss. 2016.
- D137 SCHÖTTL, F.:
Komplexität in sozio-technischen Systemen - Methodik für die komplexitätsgerechte Systemgestaltung in der Automobilproduktion.
TU München: 2015. (als Dissertation eingereicht)
- D138 BRANDT, L. S.:
Architekturgesteuerte Elektrik/Elektronik Baukastenentwicklung im Automobil
TU München: 2015. (als Dissertation eingereicht)
- D139 BAUER, W.:
Planung und Entwicklung änderungsrobuster Plattformarchitekturen
TU München: 2015. (als Dissertation eingereicht)
- D140 ELEZI, F.:
Supporting the Design of Management Control Systems In Engineering Companies from Management Cybernetics Perspective
München: TU, Diss. 2015.
- D141 BEHNCKE, F. G. H.:
Beschaffungsgerechte Produktentwicklung – Abstimmung von Produktarchitektur und Liefernetzwerk in frühen Phasen der Entwicklung
TU München: 2015. (als Dissertation eingereicht)
- D142 ÖLMEZ, M.:
Individuelle Unterstützung von Entscheidungsprozessen bei der Entwicklung innovativer Produkte.
TU München: 2016. (als Dissertation eingereicht)
- D143 SAUCKEN, C. C. v.:
Entwicklerzentrierte Hilfsmittel zum Gestalten von Nutzererlebnissen.
TU München: 2016. (als Dissertation eingereicht)
- D144 KASPEREK, D.:
Structure-based System Dynamics Analysis of Engineering Design Processes
München: TU, Diss. 2016.
- D145 LANGER, S. F.:
Kritische Änderungen in der Produktentwicklung – Analyse und Maßnahmenableitung
TU München: 2016. (als Dissertation eingereicht)

- D146 HERBERG, A. P.:
Planung und Entwicklung multifunktionaler Kernmodule in komplexen Systemarchitekturen und –
portfolios – Methodik zur Einnahme einer konsequent modulzentrierten Perspektive
TU München: 2016. (als Dissertation eingereicht)
- D147 HASHEMI FARZANEH, H.:
Bio-inspired design: Ideation in collaboration between mechanical engineers and biologists
TU München: 2016. (als Dissertation eingereicht)
- D148 HELMS, M. K.:
Biologische Publikationen als Ideengeber für das Lösen technischer Probleme in der Bionik
TU München: 2016. (als Dissertation eingereicht)
- D149 GÜRTLER, M. R.:
Situational Open Innovation – Enabling Boundary-Spanning Collaboration in Small and Medium-sized
Enterprises
TU München: 2016. (als Dissertation eingereicht)
- D150 WICKEL, M. C.:
Änderungen besser managen – Eine datenbasierte Methodik zur Analyse technischer Änderungen
TU München: 2016. (als Dissertation eingereicht)
- D151 DANILIDIS, C.:
Planungsleitfaden für die systematische Analyse und Verbesserung von Produktarchitekturen
TU München: 2016. (als Dissertation eingereicht)