A structure-based System Dynamics Approach for Assessing Engineering Design Processes

Daniel Kasperek, Markus Lindinger, Sebastian Maisenbacher, Maik Maurer

Institute of Product Development, Technische Universität München, Germany

Boltzmannstraße 15, 85478 Garching, Germany

+49 89 289 15140

daniel.kasperek@pe.mw.tum.de

Abstract: The dynamic behavior of engineering design processes is a well-known challenge within engineering. Assessing different possible process sequences for their behavior remains a major challenge within engineering design research. This paper proposes a structure-based System Dynamics approach for assessing engineering design processes for their dynamic behavior. A composition panel within the SD model is introduced to enable an eased modelling and assessment of different process sequence variants. The suggested composition panel incorporates the idea of structural methods such as Design Structure Matrix (DSM) and Multiple-Domain Matrix (MDM) into System Dynamics. By applying the DSM and MDM methods, the SD models for the different sequences become more clearly arranged and more easily to handle. Each process step within the approach is represented by the same composite concept of a rework cycle which enables the addition or deletion of process steps. This allows for a quick modeling of various variants of the engineering design process. Assessing different scenarios of engineering design process sequences by simulation offers the possibility to further improve the planning and management of engineering design processes by providing an approach to assess their dynamic system behavior.

Keywords: Engineering Design Process; System Dynamics; Design Structure Matrix; Multiple-Domain Matrix; Process Behavior

1 Introduction

The dynamic behavior of complex systems is a well-known challenge within engineering. Engineering design processes are complex systems, as a lot of factors, such as persons, resources, and iterations have to be taken into account (Smith & Morrow, 1999). Also soft factors like knowledge, motivation and skills of the involved persons play an important role within engineering design processes. (Kasperek, Maisenbacher, & Maurer, 2014; Smith & Morrow, 1999)

Considering the external influencing factors on an engineering design process as being constant, the actuating variables to influence the behavior of engineering design processes usually restrict to certain key variables such as: Size of work packages, adaption of work packages, order of process steps, persons and resources allocated to work packages. Typical questions for limiting undesired behavior are: “How can the existing resources be optimally distributed in case of a decrease of resources?” or “How can the available crew be allocated to work packages to avoid process slowdown due to possible changes within the crew?” or “How can the process be made more agile for changing customer demands?” (Kasperek, Maisenbacher, et al., 2014)

The solution space to solve these questions is an adaption or change within the structure of the process, as the reaction of a system to influencing factors is mostly caused by its underlying structure (Kasperek & Maurer, 2013; Maurer, 2007). Consequently the following questions arise: How can the structure of a system be assessed for its behavior and how can the structure of a system be optimized for a particular behavior? (Kasperek, Maisenbacher, et al., 2014)

Classically, the structure of systems is analyzed by structural modeling methods such as the Design Structure Matrix (DSM), the Domain Mapping Matrix (DMM) or the Multiple Domain Matrix (MDM) (Eppinger & Browning, 2012; Maurer, 2007). The major drawback of these tools is that they depict a static view on the system and are therefore not suitable for dynamic modeling (Diepold et al., 2010). Dynamic modeling approaches, such as System Dynamics are methods to analyze and simulate the dynamic behavior of systems on a high level of abstraction (Meier & Boßlau, 2013; Sterman, 2000). System Dynamics, though, does not offer the possibilities of dependency modeling as static aspects of systems cannot be described (Meier & Boßlau, 2013). As high level management tool, it often misses the ability to illustrate the underlying structure of the process. Therefore within this paper the combined use of structural models and System Dynamics is proposed for a structure-based System Dynamics approach for assessing engineering design processes for their dynamic behavior. The suggested approach incorporates the idea of structural methods such as DSM and MDM into System Dynamics models to allow for a better assessment of engineering design processes. By applying the DSM and MDM methods, the SD models become more clearly arranged as the relations between the particular process steps are explicitly modeled and easy to adapt within the model. Furthermore the approach serves for an eased, more modularized, SD modeling of engineering design processes.

A combination of dependency modeling and System Dynamics is especially interesting as the information needed to create structural models is available early in the development process. An integrated modeling approach would allow for an early projection of the performance of the development system over time, based on its structure. (Kasperek, Maisenbacher, et al., 2014)

The rest of the paper proceeds as follows. Section 2, presents an overview of engineering design processes and their challenges, as well as structural modeling methods and existing applications of System Dynamics for engineering design process management. In Section 3, we present a combined approach of structural modeling methods and System Dynamics for assessing engineering design processes. We demonstrate the utility of the approach with an evaluation study. Finally, the example is discussed and used to show how the proposed approach can be used as a management tool for engineering design process management.

2 Background Information

Based on an introduction into engineering design process theory, characteristics and challenges of engineering design processes from literature are given. Structural modeling is presented as a common approach for the engineering design process management and its strengths in clearly representing the underlying system structure and its weaknesses in depicting dynamic behavior are illustrated. Consequently an overview of System Dynamics in the context of engineering design process is depicted. Thereby it is focused on the composite concept of rework cycles as it offers the advantages that structural modeling is missing and otherwise.

Engineering Design Processes

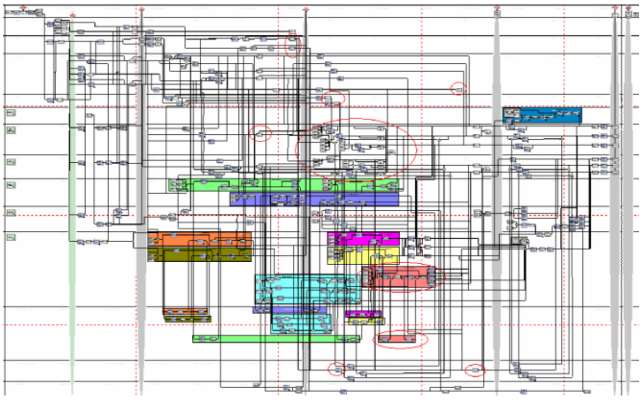


Fig. 1. Exemplary Design Process Model (taken from (Kreimeyer, Eichinger, & Lindemann, 2007))

Fig. 1 shows an exemplary engineering design process from the automotive industry. It comprises 192 different business objects that are processed by 199 process steps. It further involves seven milestones, ten organizational units and various IT systems. It becomes clear that spotting the weak point in this system is rather difficult. Engineering design processes need to be differentiated from business processes. Engineering design processes differ mainly through the uncertainty of their outcome. Therefore, they have a much less deterministic character. (Kreimeyer et al., 2007; Kreimeyer & Lindemann, 2011)

In engineering, design processes have the character of problem solving (Lindemann, 2009b), and therefore necessitate the generation of knowledge (Hatchuel & Weil, 2003; Kreimeyer, 2009). According to (Kreimeyer, 2009) engineering design process can be defined in the following way: “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.”

Iterations, during which the design is reworked, improved and refined, are typical for engineering design processes (Kreimeyer, 2009; Roelofsen, Krehmer, & Lindemann, 2008; Wynn, Eckert, & Clarkson, 2007). These iterations do not necessarily be cyclic, but can occur as leaps or loops (Badke-Schaub & Gehrlicher, 2003). Thereby leaps are characterized by forward and backward jumps in time and loops by partial sequences of process steps with reiterations to the same content (Badke-Schaub & Gehrlicher, 2003). Engineering design processes are often faced to moving targets and late changes of the concept due to the learning process during design (Kreimeyer, 2009). These aspects together cause that engineering design processes offer a low degree of repeatability (Clarkson & Eckert, 2005), and are difficult to model and plan (O’Donovan, 2004). Nevertheless, their behavior follows certain patterns concerning the product architecture, the development process and the organization and their interactions (Eppinger & Salminen, 2001). These need to be well aligned and mutually adapted to enable a smooth process (Kreimeyer, 2009). Even though there are several types of iterations depending on their originating effect, iterations in general are the typical characteristic of engineering design processes and make them difficult to manage.

The relations between the different entities of an engineering design process are the foundation of the process behavior. Therefore, the optimization of the interactions of system entities has been identified as major field of process improvement. (Flurscheim & Council, 1977; Kreimeyer, 2009; Rechtin, 1991; Wasson, 2006)

Structural modeling

Structural models are graph- or matrix-based methods used to describe the structure of complex systems and therefore often also called dependency models. One of the strengths of structural models is that they increase the system understanding by an integrated and clear illustration of the relations within complex systems. The approach of structural modelling goes back to (Steward, 1981) who introduced the Design Structure Matrix (DSM). It depicts the relations between system elements of one single domain in a square matrix. Such domains are for example components of a product or process steps to be conducted (Diepold et al., 2010). If a link exists between two elements an entry is made in the corresponding field of the matrix (Lindemann, Maurer, & Braun, 2009). DSMs can be differentiated by their field of application (Browning, 2001):

- Static DSMs: For example containing the relations between components. They mostly serve product architecture, team- based or organizational purposes.
- Time-based DSMs: For example, those are applied to problems concerning scheduling activities

DSMs are supplemented by Domain Mapping Matrices (DMMs) interrelating two domains at a time (Danilovic & Browning, 2004). In this matrix two different domains are opposed and the links in-between form the entries in the matrix (Lindemann et al., 2009). Through e.g. clustering (McCormick, 1972), these matrices allow for the analysis of the structure (Kreimeyer et al., 2007).

The third type, the Multiple-Domain Matrix covers several domains in one matrix. It consists of DSMs and DMMs, which represent different views of a system in one model, e.g. components, people, documents and requirements. If one or two domains are linked by edges of different types, several subsets of data can be considered in a Multiple-Domain Matrix (Diepold et al., 2010).

Fig. 2 presents a Multiple-Domain Matrix as an exemplary structural model. For each edge type a separate DSM or DMM can be incorporated. The approach allows for decomposing, structuring and analyzing complex systems (Lindemann et al., 2009). Additionally Fig. 2 shows how different nodes of different domains and their edges are transformed into a Multiple-Domain Matrix: As dependency models can also be described by nodes and their interrelations by edges, they are related to graph theory (Diepold et al., 2010). Edges and nodes of each matrix can be represented by a strength-based graph, which can be used to gain additional insights into the system by its visualization (Lindemann et al., 2009).

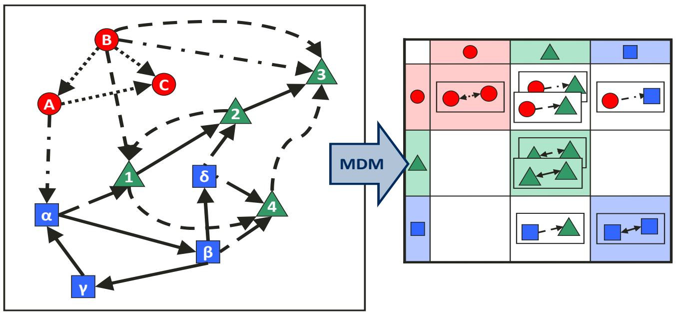


Fig. 2. Example of a structural model – the Multiple-Domain Matrix (adapted from (Lindemann et al., 2009))

According to (Lindemann et al., 2009), several analysis approaches can be applied to these models. Dependency respectively structural models can be applied not only to support decisions in engineering design concerning product structuring or organizational tasks in optimizing the alignment of process steps, but also to get a thorough understanding of a system's structure, its subunits and inherent dependencies (Diepold et al., 2010).

When discussing the dynamics of a system, there are two major points of interest. First, there is the system's evolution over time. Network theory addresses the evolution of the system structure. Network theorists developed a sophisticated toolset for predicting and simulating structural evolution (Cami & Deo, 2008). However, these approaches are often limited to graph models (Diepold et al., 2010). The other point of interest is to model the states of the same system at different points in time as DSMs, DMMs and Multiple-Domain Matrix (Eben, Biedermann, & Lindemann, 2008; Lichtenberg, Kasperek, Maisenbacher, & Maurer, 2013).

System Dynamics Modeling of Engineering Design Processes

According to (Diepold et al., 2010), structural models have the ability to show areas of dynamic behavior within the system but up to now only in a static manner. There exist already well-known tools for the modelling and simulation of dynamic behavior of systems such as System Dynamics. The major drawback of these tools is that they require deep knowledge of the process interactions and are fixed in one dynamical domain and the question of analysis-oriented transformation from the structural toward dynamic model is still in its infancy. (Diepold et al., 2010)

(Le, Wynn, & Clarkson, 2010) give an overview about System Dynamics models in engineering design. There exist already a couple of System Dynamics models of engineering design processes. Mainly they are based on a generic structure that models flows of activities between stocks that represent the current state of execution. (Ford & Sterman, 1998) introduced the rework cycle, initially developed by (Cooper, 1980), as main component of System Dynamics models in engineering. They suggest the following elements for construction of a basic System Dynamics model of an iterative product development process (Ford & Sterman, 1998):

- Process structure, including development activities and phase dependencies
- Resources
- Scope
- Targets

Through modeling the influence of rework and policies explicitly, the benefit of System Dynamics models lies in its capability to capture the dynamics and complexity of real-world systems. The framework of (Ford & Sterman, 1998) serves for improving the high-level understanding of engineering design process behavior and its impact on process performance. (Le et al., 2010)

Later authors have developed variants of the rework cycle. In Fig. 3, an adapted single phase version of the rework cycle is shown (Kasperek & Maurer, 2013). Thereby X is an index to allow for a distinct differentiation if more than one rework cycle is used.

The rework cycle operates as follows: The variable *definition X* defines the initial value of *Work Remaining X*. The work tasks then flow as a parallel flow of work with errors (amount defined by *work quality X*) accumulating into *Undiscovered Rework X*. Only if these errors are detected, rework will be necessary. The work that has to be reworked is described by the *rework discovery X* rate which processes the work to the *Work Remaining X* stock and a flow back from *Work Accomplished X* into *Work Remaining X*. An additional feedback is implemented from *Work Accomplished X* to *Work Remaining X*. The feedback is controlled by the *corrupt X* rate which is triggered by external events. The modelling construct with these two feedback loops allows to differentiate between internal rework within the phase (controlled by *rework discovery X*), such as conventional iterations during the design phase, and rework due to external events, such as the occurrence of cyclic impacts. The rework cycle finishes if the amount of accomplished work is equal to the sum of initial work to do and externally triggered work. (Kasperek, Chucholowski, Maisenbacher, Lindemann, & Maurer, 2014)

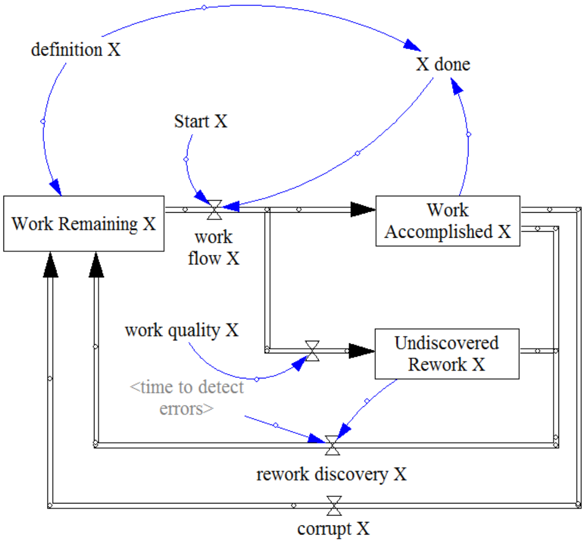


Fig. 3. Adapted single phase version of the rework cycle

The rework cycle contains various feedback loops that regulate the rates with which tasks flow between the different stocks. For example an increase of the available work may increase the work intensity which can have a negative influence on the work productivity and thus, decreases the progress rate of the overall tasks. (Kasperek & Maurer, 2013)

Rework cycle are suitable to model the typical iterations for engineering design processes. However within literature usually one single rework cycle is used to model the complete process. If the inherent sequence of process steps, or the allocation of resources to process steps is changed, these changes cannot be easily incorporated in the SD model. Therefore the assessment of different process sequences of engineering design processes remains a challenge.

3 Structure-based System Dynamics modeling of engineering design processes

To solve this challenge we propose to use several rework cycles within System Dynamics engineering design process models and a composition panel to easily change process sequences or allocations of resources between the process variants to be analyzed. For the structure-based System Dynamics approach for assessing engineering design processes, the rework cycle concept of (Kasperek, Chucholowski, et al., 2014) can be copied for each particular process step. Fig. 4 shows six rework cycles for an engineering design process with six process steps.

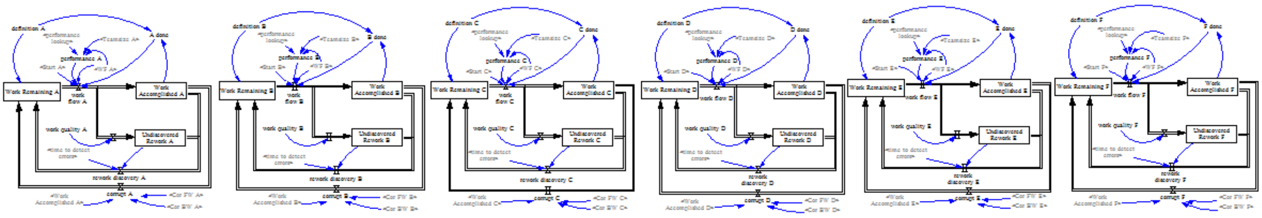


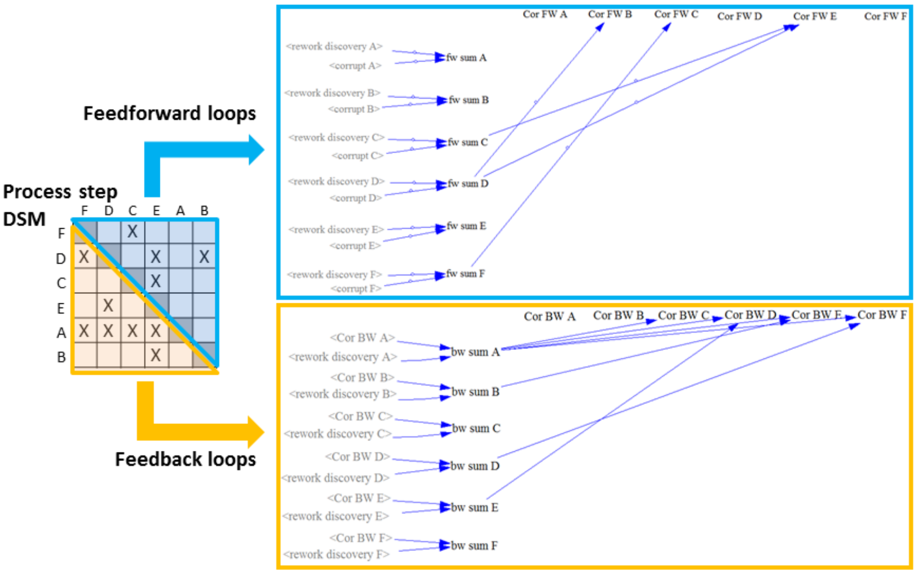
Fig. 4. Six rework cycles for an engineering design process with seven process steps

For an optimal engineering design process in the sense of shortest duration, the process sequence of the engineering design process has to be designed to generate a minimal amount of rework. Therefore it can be necessary to simulated different configurations of the process sequence. For large engineering design process with highly interconnected process steps this can result in a very complex modelling activity.

Composition panel for process sequences

To enable an eased modelling of different process sequences a composition panel is introduced. The structure of the panel is based on the structure of dependency models such as Design Structure Matrix (DSM) and Multiple-Domain Matrix (MDM).

Fig. 5 illustrates a DSM to control the engineering design process sequences and the corresponding concept for modelling the sequences within System Dynamics.


Process step DSM

	F	D	C	E	A	B
F			X			
D	X			X		X
C				X		
E		X				
A	X	X	X	X		
B				X		

Fig. 5. DSM-based composition panel for process sequences

On the left side of the illustration the process sequence DSM can be seen which forms the basis for the relations within the SD model which can be seen on the right side.

The process sequence DSM we used is built on the conventions of (Browning & Ramasesh, 2007): As illustrated in Fig. 5, the process sequence DSM is a square matrix representing the steps in a process (the shaded cells along the diagonal) and their interactions (the off-diagonal marks). One reads down an process sequence’s column to see its inputs and across its row to see its outputs (although the opposite convention is also used). For example, the process sequence DSM in Fig. 5 shows process step D providing outputs to E and B and receiving input from F. Consequently, the super-diagonal region of this DSM highlights the feed forward relationships among process steps, while its sub-diagonal region shows potential feedback loops, which imply both the need to make assumptions about unavailable inputs in upstream process steps and the potential for iteration, and thus reworks, should those assumptions prove in-adequate. (Browning & Ramasesh, 2007)

The two mentioned regions of the process step DSM are modeled in two independent parts. The variables in the DSM are thereby pointers on the corresponding variables within the rework cycles.

The feed forward relationships are modeled in the upper part. The concept *fw sum X* is an auxiliary variable that is calculated out of the variables *rework discovery X and corrupt X* which are calculated within the rework cycles. The dependencies between the *fw sums* and the *corrupt* rates depict the feedforward conditions of the engineering design process. Additionally the variable *Cor FW X* is included, to depict the amount of rework from other process steps flowing forward to the particular following process step.

The lower part represents the feedback loops within the process step sequence. The concept *bw sum X* is also an auxiliary variable that is calculated out of the variables *rework discovery X* and the shadow variable of *Cor BW X*. The dependencies between the *rework sums* and *Cor BW X* depict the feedback conditions of the engineering design process. Additionally the variable *Cor BW X* is included, to depict the amount of rework from other process steps flowing back to the particular previous process step. Correspondingly to the upper part, the mathematical equations for the dependencies can include additional conditions.

The precedence relationships are modeled in Fig 6 and, as each process step is modeled as one rework cycle, represent the starting conditions for each rework cycle (*Start A-F*). The mathematical equation for the starting condition for the particular step can also include conditions such as rates of parallelism between process steps.

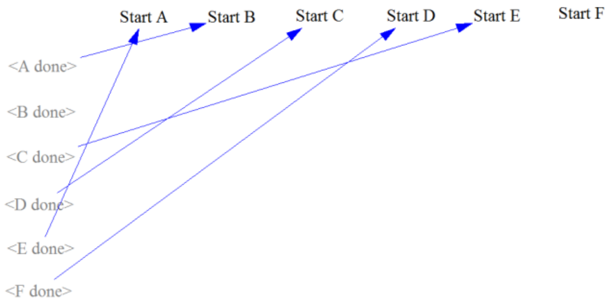


Fig. 6. Precedence relationships between the rework cycles

Fig. 7 illustrates the rework cycle concept with all integrated variables to be used in combination with the composition panel.

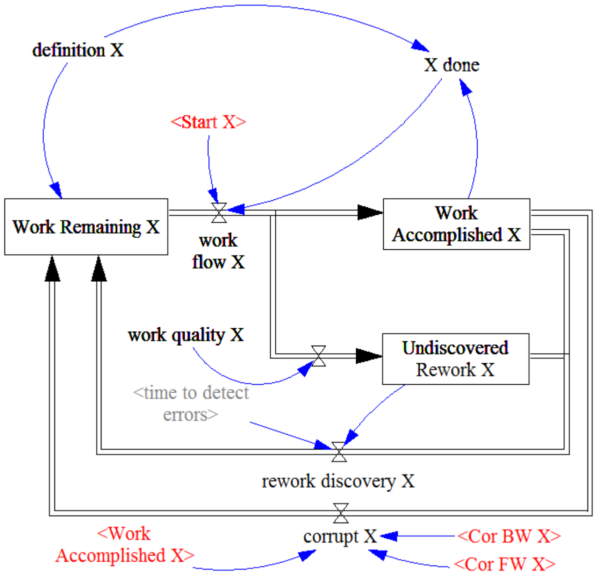


Fig. 7. Rework cycle as used in combination within the composition panel

In order to *Work Accomplished X* cannot get negative, *Work Accomplished X* is matched with the *corrupt X* rate.

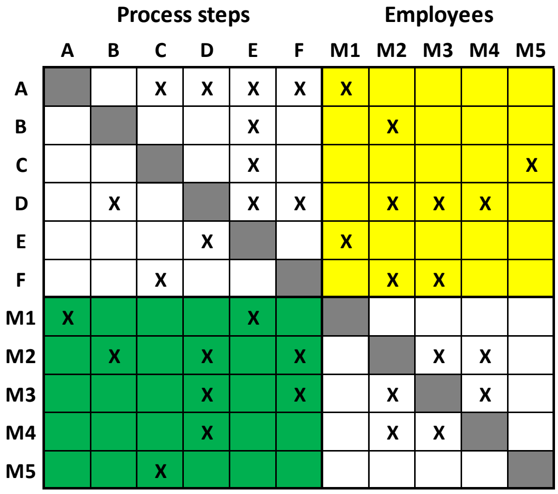
To divide the model into the rework cycles and the composition panel part offers the advantage to clearly breakdown the model. As the rework cycles are divided from the modeled relations between the process steps it offers the possibility to easily change dependencies within the model and simulate the influence on the overall process, while keeping the dynamic modeling core construct (the rework cycles) constant. This supports the accurate and quick modelling of different process configurations.

Composition panel for the allocation of resources

If different process configurations are modeled, not only the process sequence but also the allocation of resources such as employees might be changed. The abbreviations for the variables introduced within this chapter are:

Act X	auxiliary variable that is calculated out of the variables X done and *Start X*
MX	represents the employees
WF X	aggregated work flow variable for each process step depending on the amount of available employees, to avoid the necessity of allocating each individual employee to each rework cycle

The upper right subset within the MDM of Fig. 8 indicates which employees contribute to which process steps. If an employee works at more than one step at a time, the work force is assumed to be evenly distributed between the process steps.

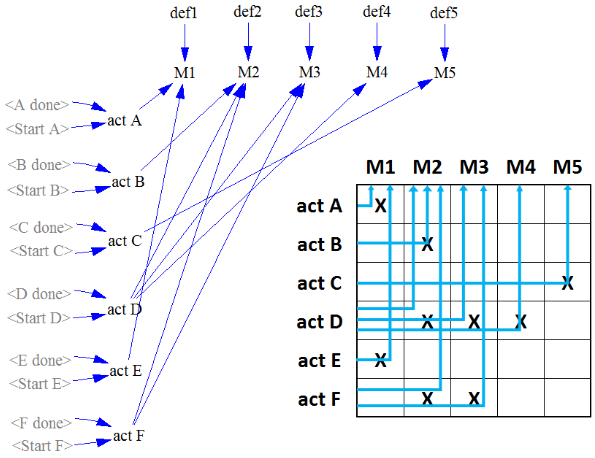


	Process steps						**Employees**				
	A	**B**	**C**	**D**	**E**	**F**	**M1**	**M2**	**M3**	**M4**	**M5**
A			X	X	X	X	X				
B					X			X			
C					X						X
D		X			X	X		X	X	X	
E				X			X				
F			X					X	X		
M1	X				X						
M2		X		X		X			X	X	
M3				X		X		X		X	
M4				X				X	X		
M5			X								

Fig. 8. Extended composition panel for the allocation of employees in MDM form

The corresponding SD model of the MDM subset is illustrated in Fig. 9.

The rows represent the process steps. *Act X* is an auxiliary variable that is calculated out of the variables *X done* and *Start X*. It indicates if the corresponding rework cycle is active at the moment. The columns represent the employees *MX*. The relations arrows from *act X* to *MX* represent which employees work at which process steps. Individual capacities of the employees can be allocated by the *defX* variable.



	M1	**M2**	**M3**	**M4**	**M5**
act A	X				
act B		X			
act C					X
act D		X	X	X	
act E	X				
act F		X	X		

Fig. 9. Allocation of employees to process steps within the composition panel

The SD concept of Fig. 9 allocates the available workforce of each employee to the active process steps. Fig. 10 illustrates the concept for the accumulation of the individually available work forces within the *WF X* variable which represents the total work force available for a process step. The *WF X* variable is introduced as aggregated variable within the rework cycle concept, to avoid the necessity of allocating each individual employee to each rework cycle. Changes within the SD concepts shown in Fig. 9 and Fig. 10 are then sufficient and the rework cycles themselves do not have to be changed.

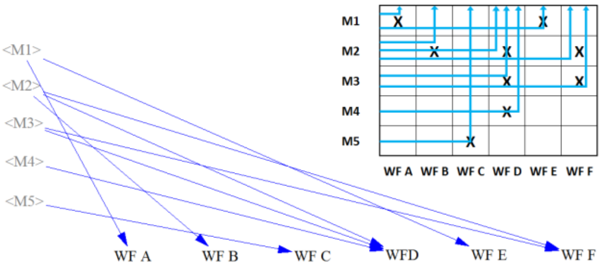


Fig. 10. Accumulation of the individually available work forces within the WF A-F variable

Fig. 11 illustrates the rework cycle concept extended for the possible allocation of resources by the composition panel.

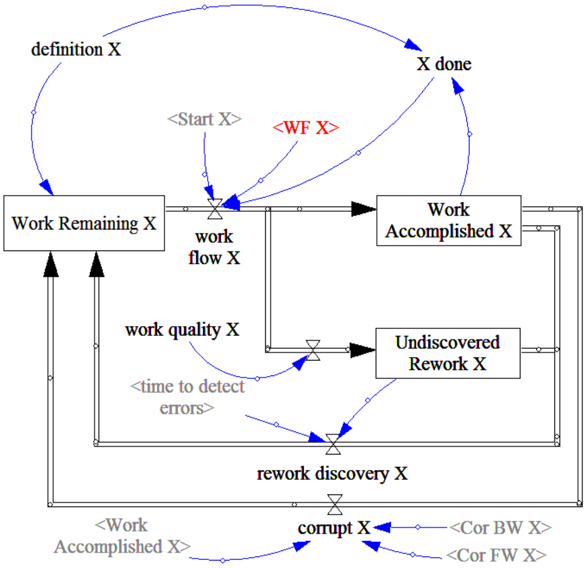


Fig. 11. Rework cycle extended for the possible allocation of resources by the composition panel.

Team performance

For the previously shown model it would be best to allocate all employees to all process steps to decrease the overall engineering design process duration. This result is usually not correct as it is known that the team size influences the overall team performance. At certain circumstances additional workers might also decrease the overall team performance.

Working in a team usually increases the communication effort to provide each team member with the necessary information. On the other hand the team performance might be higher than the summation of the individuals' performances due to synergetic effects.

Within our model the team performance is modeled as a function of the number of team members. The equation as illustrated in (1) is based on Brooks's law of software engineering which states that the maximum team size should be smaller than the square of the amount of the corresponding work indicated in man-months (Brooks, 1975). Additionally the course of the equation follows the trend for the general correlation of team size and team performance given by (Lindemann, 2009a)

$$performance = \frac{p}{MAX} \frac{MEMBERS_{process\ step\ x}}{\sqrt{WORK_{process\ step\ x}} * \left(\frac{MEMBERS^2_{process\ step\ x}}{WORK_{process\ step\ x}} + 1 \right)} \quad (1)$$

The variables are indicated in the following table:

MEMBERS	Number of team members for a process step
WORK	Initial amount of work packages for a process step (Work Remaining at time 0)
performance	Calculated correction factor of the work flow for each process step
p	Maximum amount of work performance due to synergetic effects in % (120% in our model)
MAX	Theoretically maximally possible amount of performance due to synergetic effects without saturation (implemented by a lookup table)

As synergetic effects are usually not infinite, they are limited to a certain amount of the original work flow (WF). In our model the work performance of the team can be maximally up to 20 % higher than the summation of the individuals' performances.

This is implemented by a normalization and scaling factor (p/MAX) with MAX being the maximum of the performance function. It is implemented by a lookup table. A set of the normalized performance functions for particular amounts of work packages to do can be seen in Fig. 12.

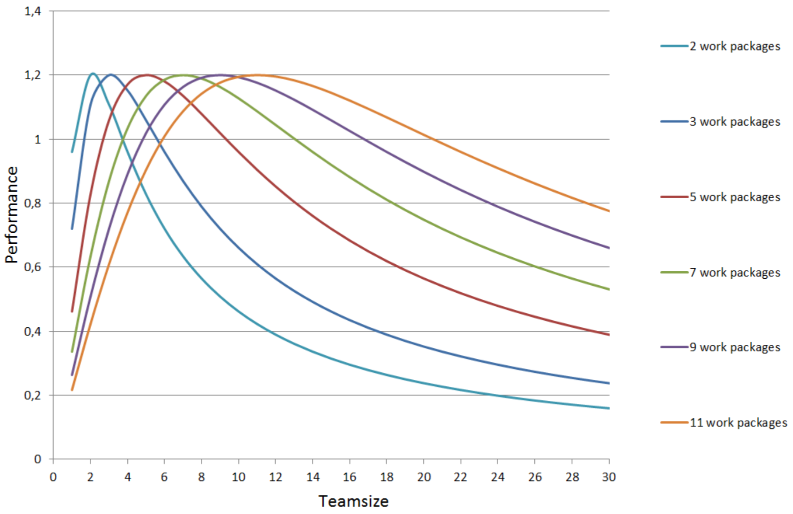


Fig. 12. Normalized performance functions for particular amounts of work packages to do

The team size dependence of the work flows is incorporated in the model by the *performance X* variable which includes the performance equation and influences the *work flows X*. The following information is necessary to calculate the rate:

- Team size (*Teamsize X*)
- Initial amount of work packages (*WF X*)
- Maximum performance depended on amount of work packages (*performance lookup*)

The corresponding rework cycle is illustrated in Fig. 13.

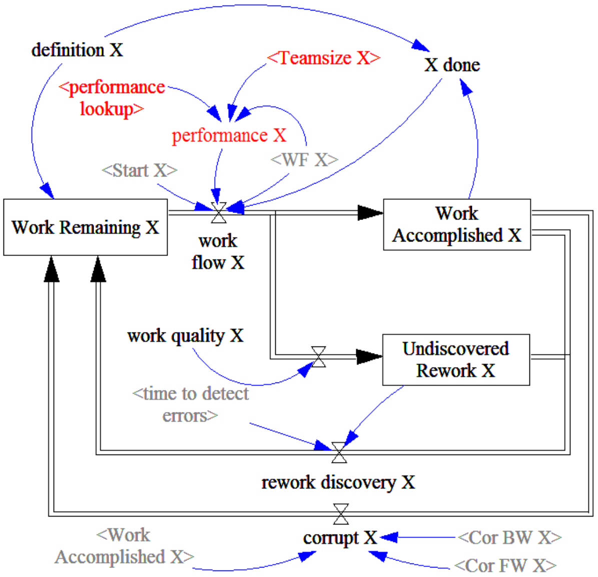


Fig. 13. Rework cycle with size dependend team performance

Project status

The overall project status is tracked by the additional *project status* variable, see Fig. 14. It compares the overall currently accomplished work within all process steps with the initial amount of overall work to do.

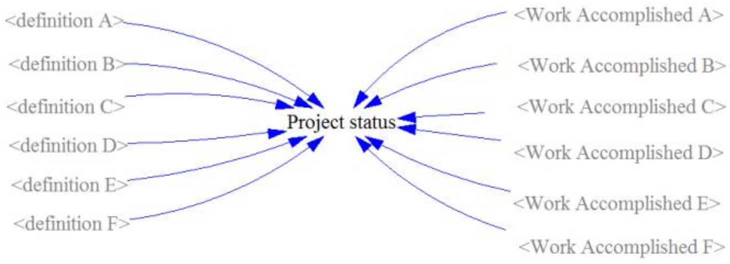


Fig. 14. Tracking of project status within SD model

The calculation of the overall project progress is an important component of the assessment of the underlying engineering design process sequences as it allows for the estimation of the durations of the particular sequences.

4 Evaluation study

To illustrate the applicability of the proposed structure-based System Dynamics approach for assessing engineering design processes a generic academic example is chosen: It represents an engineering design process with 12 process steps and 12 employees. The authors are aware of the fact that the evaluation does not prove the applicability of the approach on an industrial level, but want to show the general idea behind it and therefore choose a simple example.

From a theoretical perspective, the process duration should be the shorter the more triangularized the process sequence matrix is. A strong triangularization of DSMs is distinguished by a high degree of relations above the main diagonal of the matrix and a low degree of relations below the main diagonal. As relations below the diagonal can be interpreted as feedback loops, the degree of triangularization of the process sequence DSM can be interpreted as a measure of the existence of feedback loops. If feedback loops exist they should be short to have less impact: This corresponds to a closeness of the particular feedback relation within the matrix to the main diagonal.

For the evaluation study three different process sequences are examined. The corresponding composition panels in MDM models are illustrated within Fig. 15. It is assumed that each process has four available employees with a similar work performance. The modelled information for the employees and their allocation to process steps is kept constant for all models to allow for an assessment of the simulation results towards their theoretically correct behavior.

The three examined configurations are:

- Config 1: 6 feedback loops, which are all close to the main diagonal of the process sequence DSM
- Config 2: 12 feedback loops, which are all close to the main diagonal of the process sequence DSM

- Config 3: 14 feedback loops, which are not all close to the main diagonal

Following structural modeling theory, Config 1 should have a shorter process duration than Config 2 as the later contains the double amount of feedback loops. Config 3 should have a longer duration than Config 1 and 2 as it contains more feedback loops in total and also more "severe" feedback loops.

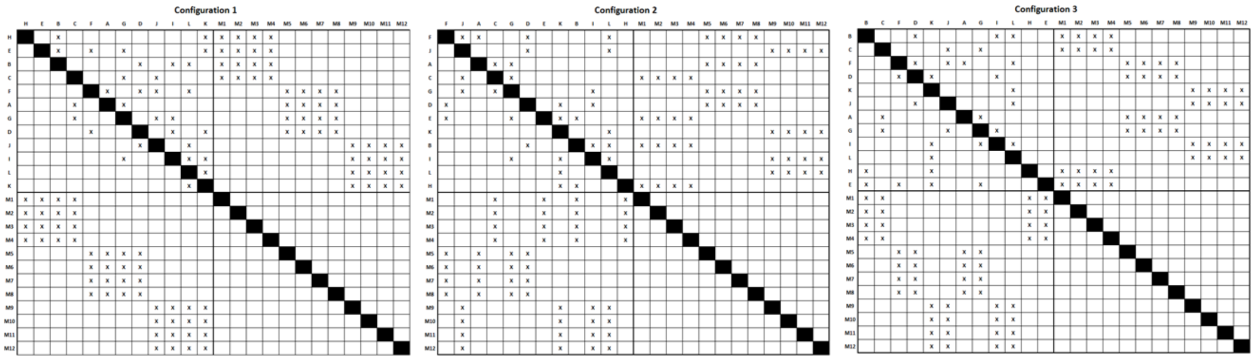


Fig. 15. Composition panels in MDM-form of the different process sequences within the evaluation study

All three configurations were modeled by the described modelling approach. The simulation results for the overall process duration are illustrated within Fig. 16.

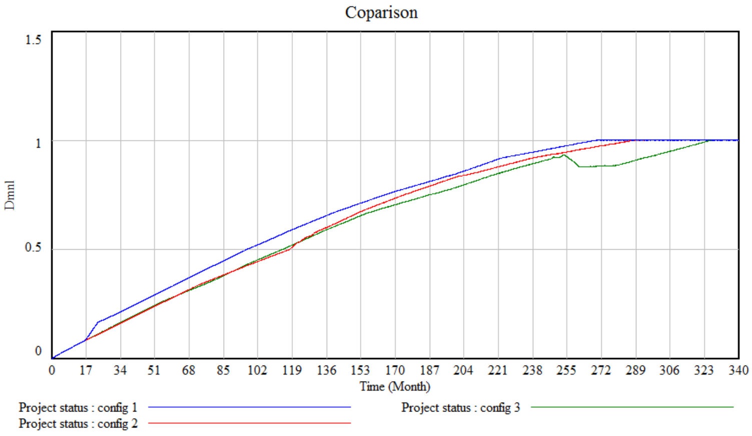


Fig. 16. Simulation results of the overall process duration for the different process sequences

The results show the expected trends from theory: Config 1 has the shortest duration. Config 3 shows a recession of the process status between 255 and 272 months, due to some "severe" feedback loops. The particular simulation results illustrated in Fig. 17 show that the "severe" feedback loops from the process step E, as illustrated in Fig. 15 are the reason for the recession.

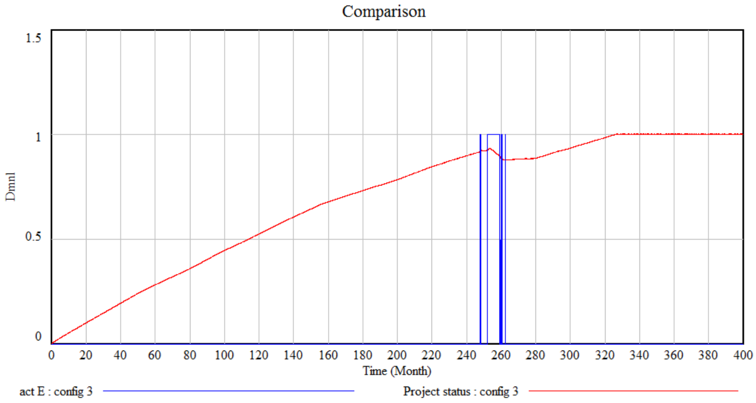


Fig. 17. Process status of config 3 in relation to rework due to feedback loops

The results show the applicability of the proposed concept for the assessment of different engineering design processes. It is able to correctly model the workflows in and between the different phases of the processes and thereby serves as an adequate approach to model different kinds of iterations within one single engineering design process model.

5 Discussion and Outlook

Even though the presented evaluation study is simplified, the application of our approach of structure-based System Dynamics models for assessing engineering design processes shows the potential of the incorporation of structural modeling methods into System Dynamics for the modeling of the behavior of different process sequences. System Dynamics is a powerful tool which is currently rarely used within engineering design. It has the potential to further develop engineering design processes by assessing for example the influence of rework and events on the overall process duration.

Not only the SD modeling of the relations within and between process steps and employees can be supported by structural methods, also other resources such as available computer, machines, or knowledge could be modeled in the same way. The domains and relation types should be adapted in a way to fit best the necessary information to build the System Dynamics model. Depending on the particular behavioral pattern to be modeled, a variety of additional domains and relation types are possible.

The modular design of the presented approach is especially helpful if various different process sequences shall be compared and assessed. As each process step is represented by the same composite concept of a rework cycle, process steps can be easily added and deleted.

The structural matrix concepts within the SD model which represent the relations between the process steps and also employees enable an easy adaption and change of these relations. This allows for a quick modeling of various variants of the engineering design process.

Even though the presented rework cycle can be reused for each process step, the parameters of each rework cycle need to be individually adapted to the particular circumstances of the process step. As for any model, the quality of the System Dynamics model is dependent on the underlying assumptions. Especially the work quality variable can significantly differ between the different phases of engineering design processes. We are currently researching on heuristics for the different parameters to be quantified within the SD model to provide further modeling support. If this information can be further improved the real system can be closer approximated.

The next step is to model and analyze different industrial engineering design processes. Additionally, the transformation from structural models into System Dynamics will be further refined. With an established System Dynamics approach at hand, it is possible to analyze several scenarios of engineering design process sequences by simulation, which will help to further improve the planning and management of engineering design processes by providing a tool to handle the dynamic system behavior.

As this publication is a result of ongoing research activities, there is more research to conduct to reach a level of industrial applicability: Currently the presented approach is applied within two industrial case studies.

6 Conclusion

This paper presents a structure-based System Dynamics approach for assessing engineering design processes. A composition panel within the SD model is introduced to enable an eased modelling and assessment of different process sequence variants. The structure of the panel is based on the structure of dependency models, such as Design Structure Matrices and Multiple-Domain Matrices.

By incorporating structural methods within the composition panel, the SD model becomes more clearly arranged as the relations between the particular process steps and the available resources are explicitly modeled and easy to adapt within the model. The modular design of the presented approach is especially helpful if various different process sequences shall be compared and assessed. As each process step is represented by the same composite concept of a rework cycle, process steps can be easily added and deleted. The structural matrix concepts within the SD model which represent the relations between the process steps themselves and also between employees and process steps enable an easy adaption and change of these relations. This allows for a quick modeling of various variants of the engineering design process. The approach offers the possibility of more easily modeling and assessing different engineering design process sequences based on the underlying structure of the processes.

Assessing different scenarios of engineering design process sequences by simulation might offer the possibility of further improving the planning and management of engineering design processes by providing an approach to model the dynamic system behavior. An exemplary case study shows the applicability of the approach as well as potential and drawbacks of the presents approach are discussed.

Acknowledgements

We thank the German Research Foundation (Deutsche Forschungsgemeinschaft – DFG) for funding this project as part of the collaborative research centre ‘Sonderforschungsbereich 768 – Managing cycles in innovation process – Integrated development of product-service systems based on technical products’.

References

Badke-Schaub, P., & Gehrlicher, A. (2003). Patterns of decisions in design: leaps, loops, cycles, sequences and meta-processes. In *Proceedings of the 14th International Conference on Engineering Design (ICED03)*. Stockholm, Sweden.

Brooks, F. P. (1975). The mythical man-month.

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), 292–305.

Browning, T. R., & Ramasesh, R. R. V. (2007). A Survey of Activity Network-Based Process Models for Managing Product Development Projects. *Production and Operations Management*, *16*(2), 217–240.

Cami, A., & Deo, N. (2008). Techniques for analyzing dynamic random graph models of web‐ like networks: An overview. *Networks*, *51*(4), 211–255.

Clarkson, J. P., & Eckert, C. C. (2005). *Design process improvement: a review of current practice* (Vol. 1). London, UK: Springer.

Cooper, K. G. (1980). Naval ship production: A claim settled and a framework built. *Interfaces*, *10*(6), 20–36.

Danilovic, M., & Browning, T. R. (2004). A formal approach for domain mapping matrices (DMM) to complement design structure matrices (DSM). In *Proceedings of the 6th international design structure matrix (DSM) workshop*. Cambridge, UK.

Diepold, K. J., Biedermann, W., Eben, K. G. M., Kortler, S., Lohmann, B., & Lindemann, U. (2010). Combining Structural Complexity Management and Hybrid Dynamical System Modelling. In *DESIGN 2010* (pp. 1045–1054).

Eben, K. G. M., Biedermann, W., & Lindemann, U. (2008). Modeling structural change over time - Requirements and first methods. In *10th International DSM Conference* (pp. 1–3). Stockholm, Sweden.

Eppinger, S. D., & Browning, T. R. (2012). *Design structure matrix methods and applications* (p. 352). Cambridge, USA: The MIT Press. Retrieved from http://www.amazon.com/Structure-Methods-Applications-Engineering-Systems/dp/0262017520

Eppinger, S. D., & Salminen, V. (2001). Patterns of product development interactions. In *International Conference on Engineering Design, ICED 01*. Glasgow, UK.

Flurscheim, C., & Council, D. (1977). *Engineering Design Interfaces: a management philosophy* (p. 138). London, UK: Design Council.

Ford, D. N., & Sterman, J. D. (1998). Dynamic modeling of product development processes. *System Dynamics Review*, *14*(1), 31–68.

Hatchuel, A., & Weil, B. (2003). A new approach of innovative design: an introduction to CK theory. *Proceedings, International Conference on Engineering Design ICED 03*.

Kasperek, D., Chucholowski, N., Maisenbacher, S., Lindemann, U., & Maurer, M. (2014). A Method for Impact Analysis of Cyclic Changes within Innovation Processes of PSS. In *submitted to: Product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems*. Windsor, Canada.

Kasperek, D., Maisenbacher, S., & Maurer, M. (2014). Structure-based Analysis of Dynamic Engineering Process Behavior. In *2014 IEEE International Systems Conference (SysCon 2014)*. Ottawa, Canada.

Kasperek, D., & Maurer, M. (2013). Coupling Structural Complexity Management and System Dynamics to represent the Dynamic Behavior of Product Development Processes. In *2013 IEEE International Systems Conference (SysCon 2013)* (pp. 414–419). Orlando, USA: IEEE.

Kreimeyer, M. (2009). *A Structural Measurement System for Engineering Design Processes*. Dr. Hut, München.

Kreimeyer, M., Eichinger, M., & Lindemann, U. (2007). Assessment of Process Networks using Graph and Network Theory Based Key Figures. *4th Future Business Technology Conference*.

Kreimeyer, M., & Lindemann, U. (2011). *Complexity Metrics in Engineering Design*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Le, H., Wynn, D. C., & Clarkson, J. P. (2010). Re-designing PD process architecture by transforming task network models into system dynamics models. *11th International Design Conference (DESIGN 2010)*, *1-3*, 319–330.

Lichtenberg, C., Kasperek, D., Maisenbacher, S., & Maurer, M. (2013). Strukturbasierte Modellierung und Bewertung von Entwicklungsprozessen von Produkt-Service Systemen. In *Tag des Systems Engineering* (pp. 1–10). Stuttgart, Germany.

Lindemann, U. (2009a). *Methodische Entwicklung technischer Produkte*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Lindemann, U. (2009b). *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden*. Berlin: Springer.

Lindemann, U., Maurer, M., & Braun, T. (2009). *Structural Complexity Management - An Approach for the Field of Product Design*. *System* (p. 240). Berlin, Germany: Springer.

Maurer, M. (2007). *Structural Awareness in Complex Product Design*. Munich, Germany: Dr. Hut.

McCormick, W. (1972). Problem decomposition and data reorganization by a clustering technique. *Operations Research*.

Meier, H., & Boßlau, M. (2013). Design and Engineering of Dynamic Business Models for Industrial Product-Service Systems. In *The Philosopher's Stone for Sustainability* (pp. 179–184). Berlin: Springer.

O'Donovan, B. (2004). *Modelling and Simulation of Engineering Design Processes*. University of Cambridge.

Rechtin, E. (1991). *Systems Architecting: creating and building complex systems* (p. 331). Englewood Cliffs, N.J: Prantice Hall.

Roelofsen, J., Krehmer, H., & Lindemann, U. (2008). Using the Design-Structure-Matrix for the Avoidance of Unnecessary Iterations. In *10th International DSM Conference*. Stockholm, Sweden.

Smith, R., & Morrow, J. A. (1999). Product development process modeling. *Design Studies*, *20*(3), 237–261.

Sterman, J. D. (2000). *Business dynamics:Systems thinking and modeling for a complex world* (p. 890). Boston: Irwin-McGraw-Hill. Retrieved from http://users.cecs.anu.edu.au/~u3951377/ENGN2225/course-files/wk02-Sterman_Feedback.pdf

Steward, D. (1981). Design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, *28*(3), 71–74.

Wasson, C. (2006). *System analysis, design, and development: Concepts, principles, and practices*.

Wynn, D. C., Eckert, C., & Clarkson, J. P. (2007). Modelling iteration in engineering design. In *ICED 2007 International Conference on Engineering Design*. Paris, France.