Carlo Leardi
Tyson R. Browning
Steven D. Eppinger
Lucía Becerril
(Eds.)

Proceedings of the 20th International Dependency and Structure Modeling (DSM) Conference

Trieste (Italy), 15 – 17 October 2018

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Herausgeber: Carlo Leardi, Tyson Browning, Steven Eppinger, Lucia Becerril

Autor: -

E-ISBN: ISBN 978-3-00-057492-4

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Table of Contents

Foreword

Welcome to the 2018 edition of the International Dependency and Structure Modeling (DSM) Conference. DSM 2018 is held on October 15th to 17th in Trieste, Italy. This year, the conference celebrates its 20th anniversary.

DSM (Dependency and Structure Modelling, also known as the Design Structure Matrix) methods have proven invaluable in designing complex systems, from product architectures to large organizations.

The International DSM Conference is the annual forum for practitioners, researchers and developers to exchange experiences, discuss new concepts and showcase results and tools. We are confident that this event will provide participants new insights, ideas, and solutions on dependency and structure modelling.

Furthermore, after last year's success we are pleased to host the second "DSM Sprint Workshop". Teams composed of a mix of researchers, practitioners, and tool providers will compete to solve one of two real industry challenges.

The papers submitted for this year's conference were each peer-reviewed by at least two members of the Scientific Committee, who made acceptance/rejection recommendations and provided helpful guidance for revisions. The accepted papers appearing in these Proceedings have been improved based on that feedback.

This volume contains 22 peer-reviewed papers that describe the recent advances and emerging challenges in DSM research and applications. They advance the DSM concepts and practice in 5 main areas:

1. Managing Risk
2. Complex Organizations
3. Product & System Architecture
4. Using Data
5. Product development

These Proceedings represent a broad overview of the state-of-the-art on the development and application of DSM. Following global trends, combining DSM Methods with data analysis, simulation and optimization is a recurring theme troughout this year's conference. Furthermore, there are a significant number of contributions with industry authors or co-authors, reflecting this balance and synergy between conceptual development and real-life industrial application, which are in the genes of the DSM Conference series.

The Program Chairs

Scientific Committee

Organizing Committee

Carlo Leardi, TetraPak, Italy
Carlo Poloni, University of Trieste & ESTECO, Italy
Prof. Steven Eppinger, Massachusetts Institute of Technology, USA
Prof. Tyson Browning, Texas Christian University, USA
Harold (Mike) Stowe, theP5DC, USA
Lucia Becerril, Technical University of Munich, Germany
Osmar Zozimo, The Journal of Modern Project Management, Brazil
Leonardo Morais de Souza, Technical University of Munich, Germany

Program Committee

All contributions in these proceedings have undergone a rigid review process. We would like to cordially thank all reviewers for their invaluable support.

Dr. Jason Bartolomei, United States Air Force, USA
Prof. Tyson Browning, Texas Christian University, USA
Ramy El Behery, Project Development
Prof. Eric Bonjour, Institut Femto-ST / Départment AS2M, France
Prof. Steven Eppinger, Massachusetts Institute of Technology, USA
Dr. Pascal Etman, Eindhoven University of Technology, Netherlands
Dr. Marija Jankovic, École Centrale Paris, France
Prof. Nitin R. Joglekar, Boston University, USA
Prof. Dieter Krause, Hamburg University of Technology, Germany
Dr. Matthias Kreimeyer, MAN Truck & Bus AG, Germany
Prof. Andrew Kusiak, The University of Iowa, USA
Prof. Udo Lindemann, Technical University of Munich, Germany
Prof. Franck Marle, École Centrale Paris, France
Prof. Carlo Poloni, University of Trieste & ESTECO, Italy
Dr. Antti Pulkkinen, Tampere University of Technology, Finland
Prof. Leonardo Santiago, Copenhagen Business School, Denmark
Dr. Venkatachalam Senthilkumar, University of Sharjah, United Arab Emirates
Dr. Kaushik Sinha, Massachusetts Institute of Technology, USA
Prof. David Wynn, The University of Auckland, New Zealand
Prof. Ali Yassine, American University of Beirut, Lebanon

The International DSM Conference is an endorsed event of the Design Society.

Part I: Managing Risk

A Failure Propagation Methodology for Analyzing Functional Models of Extremely Large Complex Systems

Leonel Akoto Chama, Oliver Bertram

Department of Safety Critical Systems & Systems Engineering, Institute of Flight Systems, German Aerospace Center (DLR), Braunschweig, Germany

Abstract: The identification of risk of potential loss of system functionality within the early stages in designing complex systems is of growing importance for risk sensitive industries. In order to enhance early design and analysis of complex system architectures using design structure matrices, this paper introduces a failure propagation index methodology for analyzing functional architecture concepts for extremely large complex systems. Unlike the classical hazard analysis techniques which become difficult to handle for extremely large complex systems, this work proposes a functional failure propagation indexing method that segments a large complex system and applies failure propagation modulating factors to estimate the criticality of the system's elements. The propagation index quantifies the relative impact of a functional failure on the overall architecture. The feasibility of the method is demonstrated using a functional model of a multifunctional actuation system architecture concept for the high-lift of a fixed wing aircraft.

Keywords: extremely large complex systems, functional failure propagation analysis, design structure matrix, system element criticality, aircraft, high lift actuation system concept

1 Introduction

The identification of risks of potential loss of system functionality during the earliest stages in designing complex systems is of growing importance (Tolga, et al., 2010). Early stage design provides the greatest opportunities to explore design alternatives and perform trade studies before costly design decisions are made. For instance, the tendency today to design the safety-critical flight control systems for multifunctionality poses many challenges (Akoto Chama, et al., 2017; Akoto Chama & Bertram, 2018). These challenges arise as a result of high safety targets and high system complexity (Sobieszczanski-Sobieski & Haftka, 1997) which may leave certain concept limitations unidentified by the designer at the early stages in development. This design process becomes even more challenging for extremely large complex systems, because classical hazard analysis techniques become more difficult to handle. On one hand, early identification and mitigation of critical design limitations are vital in designing safe and reliable large complex systems. On the other hand late identification of limitations of already established designs may require subsystem changes, which will in most cases result in changes to other parts of the subsystems, thereby increasing time and cost. Design Structure Matrix (DSM) methods (Eppinger & Browning, 2012) are widely used in generating and analyzing architectures of complex systems with a central focus on complexity management and change propagation analysis in terms of redesign or incremental development as shown in (Clarkson, et al., 2004; Giffin, et al., 2009;

Hamraz, et al., 2012; Marle & Bocquet, 2010; Fang & Marle, 2012). While these works focus on change propagation in terms of changing other subsystems in order to accommodate a change in a particular subsystem during redesign, incremental development or design for customization, they do not focus on the impact of subsystem failure on the functioning of the system (i.e. how failure of a subsystem is propagated within a complex system). Because failure of critical subsystem elements of an already established design may significantly impact the functioning of the system, their early prediction could help guide early design decisions. As a further step in enhancing the process, this work integrates preliminary safety analysis within the DSM framework by introducing a Failure Propagation Index (FPI) method for quantifying risk of potential loss of system functionality. The FPI method quantifies each functional element‘s relative failure impact on the overall architecture. The impact value is calculated from the Functional Failure Propagation Matrix (FPM) generated from the functional model. Once the distribution of the FPIs is known, valuable insights can be extracted from the architecture and early design decisions can be made to enhance or mitigate architectural concept limitations. Unlike in the aforementioned works by (Clarkson, et al., 2004; Giffin, et al., 2009; Hamraz, et al., 2012; Marle & Bocquet, 2010; Fang & Marle, 2012) which focus on how change is propagated during engineering design change, the proposed method in this work focuses on predicting how a failure of a particular subsystem element prevents other subsystems from performing their intended functions.

2 Conventional Design Approach

Typically the design of systems begins with stakeholder analysis, then requirements analysis and ends with an architecture as shown in Figure 1. Of particular importance in the process is the analysis and allocation of functions carried out after the requirements analysis and before the synthesis of the concept.

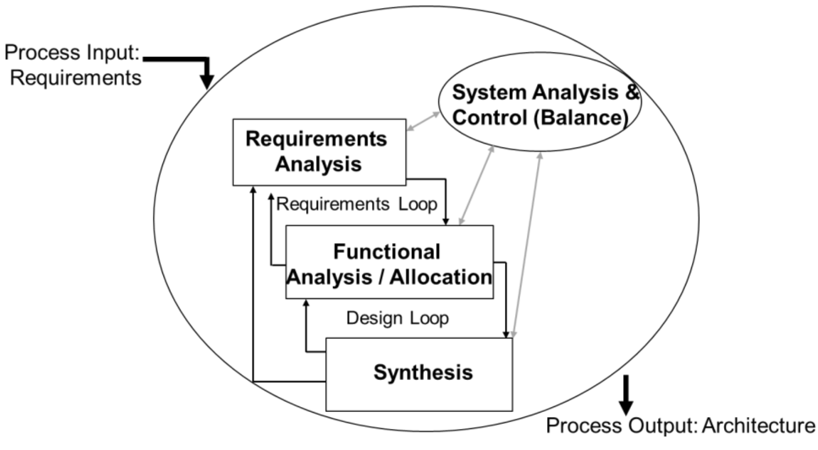


Figure 1. Classical design process

At this stage, complex systems pose many design challenges and even greater challenges for extremely large complex systems, thus methods which are also applicable to functional networks have been developed to manage system complexity and change propagation (Hamraz, Caldwell, & Clarkson, 2012; Clarkson, Simons, & Eckert, 2004; Fang & Marle, 2012; Carlos Inaki, 1998; Thebeau, 2001). Also, within this stage, the classical hazard analysis techniques such as the Functional Hazard Analysis (Clifton A, 2005; Dalton, 1996) are performed on the system to identify potential hazardous elements in the design. While these hazard analysis techniques are sufficient, they become very difficult to apply for extremely large complex systems which may lead to potentially unidentified hazardous system elements. Thus, if design flaws are not identified and mitigated early enough, this may result in costly design changes later in the design process or even catastrophic failures during the operational phase of the system (Akoto Chama & Bertram, 2018).

3 Method

Every system is fundamentally made up of functional elements. How these elements are connected and interact with each other defines the functional architecture of the system. Understanding how these functions affect each other and how they work together to accomplish the mission of the system is vital in creating optimal system architectures. In order to identify critical system elements whose failure impact can greatly affect the functioning of the system, this work proposes a three step FPI approach with main focus on the functional failure propagation analysis that quantifies the relative failure impact of subsystem elements. The FPI method introduced in this work, has been developed to enhance the design process by reducing the risk of design flaws propagated to later stages in the design process for extremely large complex systems. The principle of the method is explained below.

3.1 Functional Model (Step 1)

The process begins by creating a functional model of the system to be designed. A functional model of a system is an abstraction that represents the system's functions and their interactions (Akoto Chama, et al., 2017; Stone & Wood, 2000; Hutcheson, et al., 2007; Chakrabarti, et al., 2011; Pahl, et al., 2007). It represents the transformation of energy, material or signal information flows as they pass through the system elements. It defines how the functions will operate together to perform the system mission. Generally, more than one functional model can satisfy the system requirements and thus a suitable functional model depends on the level of required detail that should be analyzed. In order to explain the proposed method, consider the arbitrary seven element system as shown in Figure 2. It is assumed that the elements of the system are functions, which are connected to each other as shown.

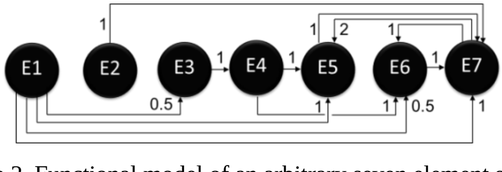


Figure 2. Functional model of an arbitrary seven element system

3.2 Design Structure Matrix (Step 2)

The DSM, also referred to as dependency structure matrix is a tool for network modeling (Eppinger & Browning, 2012). It is used to represent a system's architecture (or design structure) by mapping the interactions among the elements that make up a system. The DSM is represented as a square $\widetilde{N} \times \widetilde{N}$ matrix, with relations (or interactions) among the set $\widetilde{N}$ of system elements. One can think of a DSM as a collection of cells (e.g. E1 to E7 in Table 1) along the diagonal of the matrix as representing the system elements (Figure 2) analogous to the nodes in the digraph model (Eppinger & Browning, 2012). The diagonal cell has inputs entering from its left and right sides and outputs leaving to above and below as shown on Table 1. The corresponding marks in the off-diagonal cells indicate the sources and destinations of the inputs and outputs, analogous to the directional arcs in a digraph. The inputs to an element in a row (which are outputs of other elements) are indicated by marks in that row. The outputs from an element in a column (which are inputs to other elements) are indicated by marks in that column. For the seven functional elements system shown above, the corresponding DSM is represented as shown on Table 1.

Table 1. Design structure matrix of the functional model above

		E1	E2	E3	E4	E5	E6	E7
Element 1	E1	E1						
Element 2	E2		E2					
Element 3	E3	x		E3				
Element 4	E4			x	E4			
Element 5	E5	x			x	E5		x
Element 6	E6	x			x		E6	x
Element 7	E7	x	x			x	x	E7

3.3 Functional Failure Propagation Analysis (Step 3)

If a functional element fails, it is possible that other elements (functions) within the functional network are affected, synonymous to change propagation in (Clarkson, et al., 2004; Giffin, et al., 2009; Hamraz, et al., 2012; Marle & Bocquet, 2010) . In this section, all functions which are affected as a result of the failure of a particular function are captured. The Functional Failure Propagation Analysis (FPA) generates the information on how functional failures are propagated within the functional (network) model. For the propagation analysis the following definitions are used:

Downstream elements: Elements along the affected paths to which the output of the element under consideration goes.
Upstream elements: Elements along the affected path from which the element under consideration receives inputs.
Modulation: The change in the effect of a failure as it is propagated within the network.
For example, consider the seven element functional model as shown in Figure 2 and let the element E1 be degraded or fail. Then, there are four different failure propagation possibilities (or scenarios) within the internal network structure;

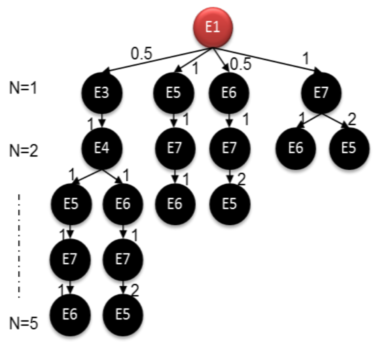
Case 1: The failure is not propagated
Case 2: The failure is propagated equally upstream and downstream across the entire network without modulation.
Case 3: The failure is propagated upstream and downstream across the entire network with modulation.
Case 4: The failure is propagated only through certain elements within the network, with or without modulation.

For extremely large complex systems, the fourth case becomes extremely challenging to handle since the elements affected must be identified for the analysis. In this work a generalized approach for capturing failure impact is presented. The failure propagation is captured within a predictive matrix called Failure Propagation Matrices (FPMs) which are DSMs whose off-diagonal cell entries represent the propagation paths and the magnitude of the effect on an element within the propagation path. The magnitude of the effect is reflected in the strength of the connection (0.5 for weak, 1 for medium and 2 for strong) which defines the relative importance of the connection within the functional network. The importance of the connection is based on its necessity for efficient system operation. A basic sensitivity analysis showed that the relative criticalities of the system elements were mostly stable to small changes in the connection strengths. Also the term "predictive" is used in describing the FPMs because the entries are based on subjective judgement and tied closely to the intended behavior of the subsystem within the system.

The propagation of the failure across different elements may differ according to the four possibilities listed above. Figure 3, shows scenarios 2 and 3 (Cases 2 and 3), where the entire network is affected, is affected. Figure 3(a) shows a failed element within the DSM and Figure 3(b) shows a tree representation of the elements affected as a result of this failure. The tree is generated using a depth first search algorithm beginning from element E1, and representing all possible paths; hence multiple elements appear in different tree branches.

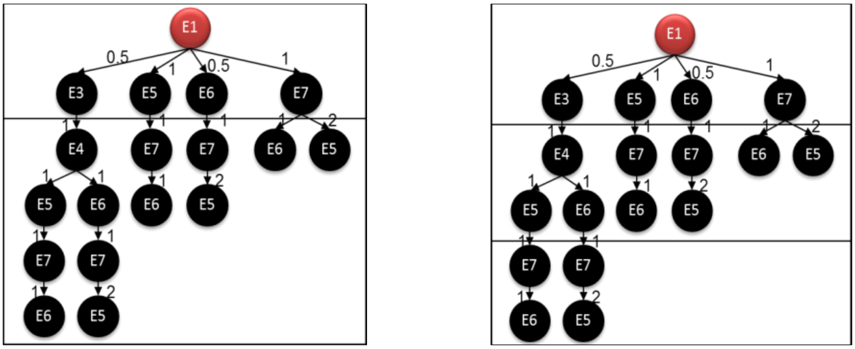
	E1	E2	E3	E4	E5	E6	E7
E1	E1						
E2		E2					
E3	0.5		E3				
E4			1	E4			
E5	1			1	E5		2
E6	0.5			1		E6	1
E7	1	1			1	1	E7

(a) E1 failed element



(b) Depth first search tree

Figure 3. An exemplary failed element and its corresponding tree



(a) Two segment partition (b) Multiple segment partition

Figure 4. Segmentation of failure propagation paths

Since E1 has no upstream elements, they are not shown. Furthermore, in order to capture propagation effect across the entire network according to scenario 3 (Case 3), the tree in Figure 3 (b) can be segmented as shown in Figure 4. The tree elements can be partitioned into two segments as shown in 4(a) or multiple segments as shown in 4(b) according to impact of the failure on them. The calculated value from the Failure Propagation Matrix (FPM) is called the Failure Propagation Index (FPI) and determines the relative criticality of the element. To obtain an element's FPI, let i and j be elements of the functional model such that when element i fails, element j is affected (i.e. j is an element that belongs to the tree generated from element i). Also let $e_{i,j}$ be the edge preceding element j, along the tree path between i and j. For a given depth N, let $S(i,N)$ be the set of elements j (in the tree of i) at depth N from i. Note from Figure 3 that the same element j may appear at different depths. Then the summed up edge magnitudes at a given depth N are:

$$P(i,N) = \sum_{j \in S(i,N)} e_{i,j} \qquad (1)$$

Each depth is assigned a modulation factor M(N). The modulation factor is chosen to reflect how an element failure may impact other elements within the network. For example, for a functional network which is designed such that distant elements are less affected, a modulation factor which is inversely proportional to the distance can be chosen. Thus, the FPI for i is the following sum:

$$FPI_i = \sum_N M(N)P(i,N) \quad (2)$$

If $\left(M(N)\right)_{N=1\ldots MaxN}$ and $\left(P(i,N)\right)_{N=1\ldots MaxN}$ are interpreted as vectors, then the FPI for i is the inner product of these two vectors:

$$FPI_i = \langle \left(M(N)\right)_{N=1\ldots MaxN}, \left(P(i,N)\right)_{N=1\ldots MaxN} \rangle \qquad (3)$$

Since failure propagation may be different for upstream and downstream elements, introducing direction on (3) yields:

$$FPI_i^- = \langle \left(M_{up}(N)\right)_{N=1\ldots MaxN}, \left(P_{up}(i,N)\right)_{N=1\ldots MaxN} \rangle \qquad (4a)$$

$$FPI_i^+ = \langle \left(M_{dn}(N)\right)_{N=1\ldots MaxN}, \left(P_{dn}(i,N)\right)_{N=1\ldots MaxN} \rangle \qquad (4b)$$

Equation (4a) gives the calculated partial FPI (FPI_i^-) from elements affected upstream while equation (4b) gives the calculated partial index (FPI_i^+) from elements affected downstream. The multiplicative factor $M_{up}(N)$ gives the upstream dependency modulating factor as a function of the distance N. Similarly the downstream dependency modulating factor is given by $M_{dn}(N)$. The sum of the upstream and downstream partial FPIs ($FPI_i^- + FPI_i^+$), gives the FPI of the element under consideration. In matrix form, equation (4) can be written as shown in equation (5). Equation (5) represents the Frobenius inner product of the Modulation Matrix and the Propagation Matrix

$$FPI_i^{\pm} \langle \begin{bmatrix} M_{up}(1) & M_{up}(2) \ldots M_{up}(MaxN) \\ M_{dn}(1) & M_{dn}(2) \ldots M_{dn}(MaxN) \end{bmatrix}, \begin{bmatrix} P_{up}(i,1) & P_{up}(i,2) \ldots P_{up}(i,MaxN) \\ P_{dn}(i,1) & P_{dn}(i,2) \ldots P_{dn}(i,MaxN) \end{bmatrix} \rangle \qquad (5)$$

For compactness, equation (5) can be written as shown in equation (6) where M represents the Modulation Matrix and P represents the Propagation Matrix.

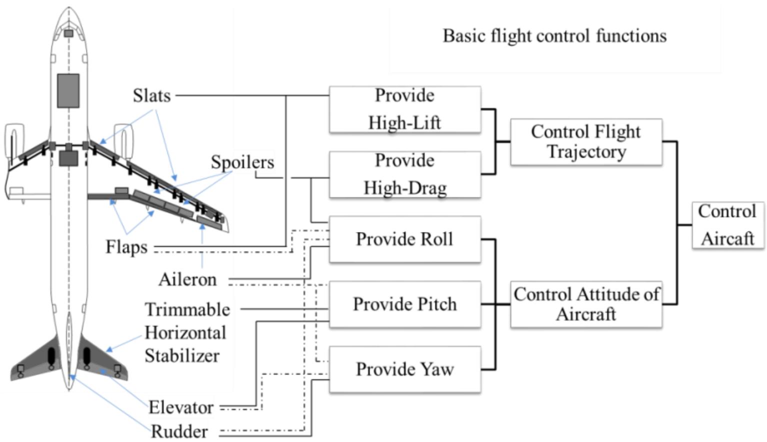
$$FPI_i^{\pm} = \langle M, P(i) \rangle \qquad (6)$$

For a tree with partitions (e.g. Figure 4), elements within the same partition can be modulated similarly while elements belonging to different partitions can be modulated differently. Such modulation is chosen to reflect potential impact of element failure on other elements (e.g. see application case in next section). In case of failure, the FPI of an element reflects the number of elements affected within the network structure of the functional model and the severity of impact. A high FPI value can be as a result of lots of affected elements with low severity or a few affected elements with high severity. Capturing this information early in the design process can be vital in optimal module formation within the function allocation stage or in making critical design decisions.

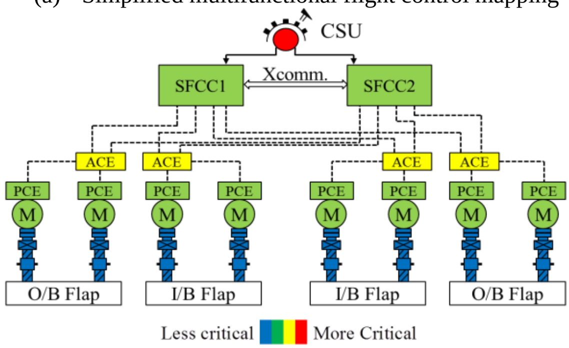
Note: The results of the analysis are influenced by the chosen values for edge (connection) strengths in the network and the modulation factors. Thus care must be taken in choosing these parameters and the resulting observations must be analyzed accordingly.

4 Application on a Multifunctional Flap Actuation Concept

Multifunctionality in flight control system presents many advantages for efficient flight which leads to reduction in fuel burn (Akoto Chama & Bertram, 2018; Akoto Chama, et al., 2017; Reckzeh, 2014; Cook & de Castro, 2004; Reckzeh, et al., 2012).



(a) Simplified multifunctional flight control mapping



(b) An exemplary multifunctional flight control flap actuation system (first two O/B and I/B flaps for the left wing and the last two for the right wing)

Figure 5. Multifunctional flight control surfaces and actuation system

ACE	: Actuator Control Electronics	O/B Flap	: Outboard Flap
CSU	: Command Sensor Unit	PCE	: Power Control Electronics
I/B Flap	: Inboard Flap	SFCC	: Slat/Flat Control Computer
M	: Motor	Xcomm	: Communication Signal

Figure 5(a) shows a simplified example of the mapping between the control surfaces on the aircraft (right) and flight control functions (left). The solid lines show the classical mapping while the dashed lines show possible functionalities that could be added to the control surfaces. The underlying actuation systems that actuate the control surfaces are very complex and present many design challenges. Thus, in order to demonstrate the proposed methodology, this paper analyses the functional network of the fully distributed flap actuation system concept (Recksiek, 2009) as an application. The physical layout and possible criticality distribution of the analyzed actuation system is shown in Figure 5(b). The design problem was to design a flap actuation system that allows the flaps to perform multiple functions such as increasing the maximum lift coefficient, spanwise lift distribution as well as roll assist. Applying the three step approach described above, a functional model was created composing of 140 interconnected elements. A DSM was then created and a functional failure propagation analysis was performed. Since it was assumed that there was no information about the type of physical system solution, the analysis was based only on analyzing the functional network structure. Here the tree was partitioned into 2 sections as shown in Figure 4(a).

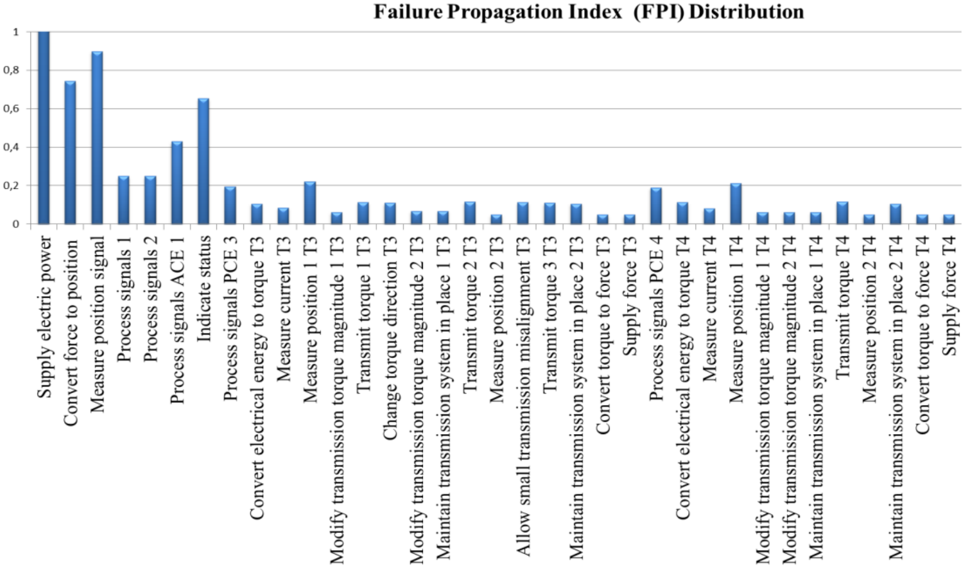


Figure 6. FPI distribution of a section of the functional architecture

The upper segment contained the failed element and elements that are directly affected by the failure without modulation. The lower segment contained elements that are indirectly affected by the failure with a distance dependent modulation of $M_{dn}(N) = 1/(2N)^2$ and $M_{up}(N) = 1/(4N)^2$ for the downstream and upstream elements respectively.

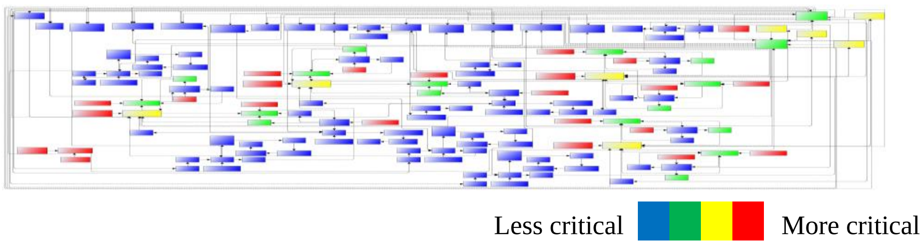


Figure 7 FPI color distribution of the complete functional model

These modulation factors were chosen in this way because it was assumed that the greater the distance between elements the lesser their dependency on each other and also that dependency for upstream elements reduces faster than that of downstream elements. For this work all connection strengths where chosen as unity for preliminary analysis and because no information was assumed for the technical solution of the functional model. Figure 6 shows a section of the FPI distribution for the elements in the model. For symmetry purposes, only a section of one wing of the entire architecture is shown. Figure 7 shows the color distribution of the complete functional model of the flap actuation system. This color distribution was created according to the magnitude of the FPI generated using equation (4), with normalized distribution as follows: $Blue \leq 0.2$, $0.2 < Green \leq 0.4$, $0.4 < Yellow \leq 0.6$ and $0.6 < Red$. As expected, the distribution showed that more highly connected elements are more critical than less connected elements. Because the FPI of the electric power supply is extremely high, a potential early design decision would be to introduce a second power supply. Also, another enhancement decision could be to introduce redundant “Process Signals ACE” function due to the high FPI. The general observation made from the functional architecture is that highly connected elements with many downstream elements are more critical than those that are less connected with fewer downstream elements. Also functional elements with electrical inclined solutions (e.g. Supply Electric Power) which are highly connected have higher criticality values than those which are not (e.g. Measure Position 2 T4). On the other hand the two green boxes on the top right of Figure 7 represent the SFCCs, though highly connected, they are less critical because of redundancy. This equally shows the effect of redundancy using the FPI method.

4 Conclusion and Further Recommendations

Designing extremely large complex systems is a challenging task, especially when dealing with hundreds or thousands of interacting elements. This makes it difficult to identify high risk elements without prior knowledge in the design. As shown in this work, the Failure Propagation Index can therefore be used as a tool to help with the identification of such elements which may otherwise go unnoticed by the designer. If such elements are not identified early in the design process, this may lead to costly design changes or even catastrophic failures in the operational phase. The index formulation presented in this work serves as a systematic way to identify high risk elements in order to improve on the initial concept. The index can either be applied within the concept development phase, as shown for the functional model of the multifunctional actuation concept or for assessing existing design concepts. The FPI methodology uses segmentation of extremely large complex systems and modulation for modulating the failure as it is propagated within the network structure. The FPI method determines the internal impact of the element failure, by capturing its effect within a particular architecture network. With this method, only element connections, connection weights, and distance weights are needed to capture preliminary system element failure impact during concept generation. This aspect is especially useful for extremely large complex systems because of the huge challenges in capturing and processing their complex system behavior. Though this method can be used as a first step in understanding element network impact for extremely large complex systems, a possible enhancement could be by modifying the modulation factor to depend not only on the distance but also on the failed element. Another enhancement could be by introducing multiple connections of different types between system elements, which gives more detail to the functional network and thus, allows the possibility for further analysis on the system architecture. Nevertheless, as a first step in capturing element failure impact, the FPI method has been successfully demonstrated.

References

Akoto Chama, L., & Bertram, O. (2018). Identifying Enhancement Potentials of Actuation Systems Using a Criticality Index. 8th International Conference on Recent Advances in Aerospace Actuation Systems and Components. Toulouse, France: 8th International Conference on Recent Advances in Aerospace Actuation Systems and Components,INSA.

Akoto Chama, L., Bertram, O., & Schumann, H. (2017). Generation of Potential System Architectures by Applying a Stochastic Clustering Algorithm in the High-Lift Actuation Preliminary Design Process. 19th International dependency and structure modeling conference, DSM 2017. Espoo, Finland: 19th International Dependency and Structure Modeling Conference, DSM 2017.

Carlos Inaki, G. F. (1998). Intregration analysis of product architecture to support effective team co-location. Massachusetts: Massachusetts Institute of technology.

Chakrabarti, A., Shea, K., Stone, R., Cagan, J., Campbell, M., Hernandez, N. V., & Wood, K. L. (2011). Computer-Based Design Synthesis Research: An Overview. Journal of Computing and Information Science in Engineering, 11.

Clarkson, P. J., Simons, C., & Eckert, C. (2004). Predicting Change Propagation in Complex Design. Transactions of the ASME, Journal of Mechanical Engineering, 126, 788-797.

Clifton A, E. (2005). Hazard Analysis Techniques for System Safety. New Jersey: John Wiley & Sons, Inc.

Cook, M. V., & de Castro, H. V. (2004). The Longitudenal Flying Qualities of a Blended-Wing-Body Civil Transport Aircraft. The Aeronautical Journal, 75-84.

Dalton, J. (1996). Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment - ARP4761. Pennsylvania: SAE - International; The Engineering Society for Advanced Mobility Land Sea Air and Space.

Eppinger, S. D., & Browning, T. R. (2012). Design Structure Matrix Methods and Applications. Cambridge, Massachusetts: The MIT Press.

Fang, C., & Marle, F. (2012). Dealing with Project Complexity by Matrix-Based Propagation Modelling for Project Risk Analysis. Journal of Engineering Design, Taylor & Francis.

Giffin, M., de Weck, O., Bounova, G., Keller, R., Eckert, C., & Clarkson, P. J. (2009). Change Propagation Analysis in Complex Techmical Systems. Transactions of the ASME, Journal of Mechanical Design, 131.

Hamraz, B., Caldwell, N. H., & Clarkson, P. J. (2012). A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design . Transaction of the ASME, Journal of Mechanical Design, 134.

Hutcheson, R. S., McAdams, D. A., Stone, R. B., & Tumer, I. Y. (2007). Function-Based Systems Engineering (FUSE). Paris: International Conference on Engineering Design, ICE'07.

Marle, F. V.-A., & Bocquet, J.-C. (2010). Interaction-Based Risk Clustering Methodologies and Algorithms for Complex Project Management. International Journal of Production Economics, 142, 225-234.

Pahl, G., Beitz, W., Feldhusen, J., & Grote, K. H. (2007). Engineering Design, A Systematic Approach (Third Edition ed.). London: Springer-Verlag.

Recksiek, M. (2009). Advanced High Lift System Architecture with Distributed Electrical Flap Actuation. Workshop on Aviation System Technology. Hamburg: Technische Universität Hamburg-Harburg.

Reckzeh, D. (2014). Multifunctional Wing Moveables: Design of the A350XWB and the Way to the Future. 29th Congress of the International Council of the Aeronautical Sciences. St. Petersburg, Russia: ICAS.

Reckzeh, D., Bernhard, S., Andreani, L., & Sutcliffe, M. (2012). Patentnr. US 8,336,829 B2. United States of America.

Sobieszczanski-Sobieski, J., & Haftka, R. T. (1997). Multidisciplinary Aerospace Design Optimization Survey of Recent Developments. Springer-Verlag.

Stone, R. B., & Wood, K. L. (2000). Development of a Functional Basis for Design. Journal of Mechanical Engineering, 122, 359-369.

Thebeau, R. E. (2001). Knowledge management of system interfaces and interactions for product development. Massachusetts : Massachusetts Institute of Technology.

Tolga, K., Irem Y, T., & Jensen, D. C. (2010). A Functional Failure Reasoning Methodology for Evaluation of Conceptual System Architectures. Springer-Verlag, 209-234.

Contact: L. Akoto Chama, German Aerospace Center (DLR), Safety Critical Systems & Systems Engineering, Lilienthalplatz 7, 38108, Braunschweig, Germany, Tel: +49 531 295-2841, Fax: +49 531 295-2647, e-mail: leonel.akoto@dlr.de

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Using MDM and random walk for analyzing the combined influencing strength of Risk-DSM

XingQi Zou, Qing Yang

School of Economics and Management, University of Science & Technology Beijing

Abstract: In the complex R&D process, changes from function and component may cause uncertainties. To solve the problem, the paper builds Multi-domain Matrix (MDM) of "function-component-risk" to identify risk factors and its potential relationship. Taking the results of MDM as input, the paper uses random walk algorithm to analyze the influencing strength between different risk factors. Further, the paper calculates the combined influencing strength based on direct and indirect risk propagation. An industrial example is provided to illustrate the proposed model. Results indicate that the change of function and component can discover the risk factors and its potential relationship, and the indirect influencing is very important when measuring the combined influencing strength.

Keywords: Multi-domain Matrix (*MDM*)*, random walk algorithm, change propagation, Risk- DSM*

1 Introduction

R&D project is a complex system involving project, process and organization management, complexity and uncertainty are the most prominent feature (Yang et al., 2015). As the primary source of uncertainty of project, complexity has been extensively explored in the literature (Qazi et al., 2016). The uncertainty will produce additional costs and affect project performance if managers fail to address it (Shenhar, 2001). Moreover, the complexity and uncertainty derive principally from its sophisticated function and multitudinous components of projects (Eckert et al., 2016; Koh et al., 2012). Therefore, the change of function and components will bring high risk, which dramatically increases the difficulty of project management (Ackermann et al., 2014).

So, to identify risks in R&D projects and determine the relationship between different risk factors, we present an innovative approach to analyze the risks using extend-MDM (E-MDM). The paper has three key contributions to practice: 1) it presents the "function-component-risk" E-MDM to identify risk factors and determine its potential relationship; 2) taking the analyzing results of MDM as input, the paper uses random walk algorithm to calculate the influencing strength; 3) the paper analyzes the influencing strength based on direct and indirect risk propagation and then calculates the combined influencing strength through all possible propagation paths.

2 The calculation of initial Risk-DSM based on MDM

The DSM proposed by Steward (1981) is a powerful structural method to represent the elements comprising a system and their dependencies (Yang et al.,2015).The MDM is an extension of DSM modeling in which two or more DSM models in different domains are represented simultaneously, each single-domain DSM is on the diagonal of the MDM, and the off-diagonal blocks are Domain Mapping Matrix (DMM) (Eppinger & Browning, 2012). The DMM is a (typically) non-square matrix mapping the domain of one DSM to the domain of another DSM (Eppinger & Browning, 2012).

In this paper, the Risk-DSM (R_DSM) implements the risk factors involved in R&D project and the relationship between different risk factors. We builds upon the E-MDM of "function-component-risk" to determine the R_DSM and analyze the risk factors. As shown in figure 1, the MDM consists of three essential parts: functional DSM (F_DSM), component DSM (C_DSM) and the risk DMM (R_DMM).

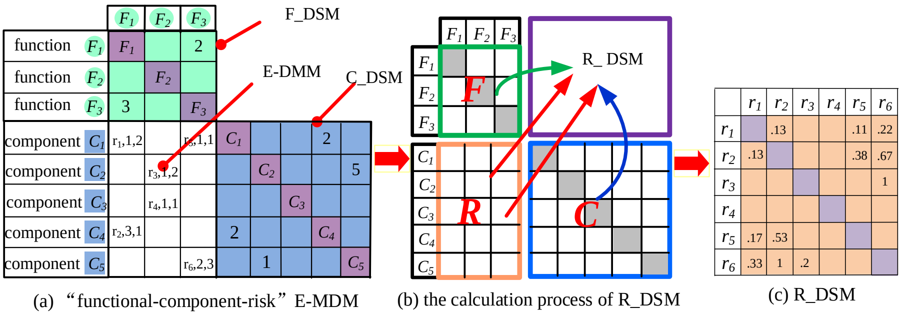


Figure 1. The calculation of initial R_DSM

The model of F_DSM/ C_DSM describes the functional/componential changes and its relationship in R&D projects; the R_DMM reflects the risk factors arising from functional or component changes and shows the relationship between functional/componential changes and risk factors. It can be seen from the R_DMM that risk may be caused by functional or componential changes, and the traditional DMM cannot describe these two changing relationships.

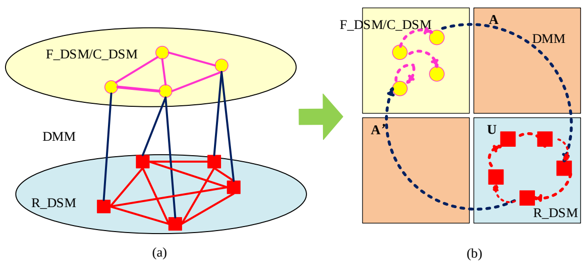
So, the paper builds the E-DMM, as shown in figure 1, each element in E-DMM can contain three parts, $E-DMM(r_i,\alpha,\beta)$ r_i is the risk factor, α is the impact relationship between risk and functional changes (Column) and β is between risk and component changes (row). For instance, $(r_1,1,2)$ reflects the degree of risk factors are affected by functional changes (F_1) is 1, and affected by componential changes (C_1) is 2.The paper uses the value ranging from 1 to 5 to quantify the intensity of relationship. The higher the value, the stronger the impact relationship.

3. Using random walk to calculate the influencing strength between different risk factors

The random walk method, a recent innovation, can be used to deduce the influencing strength. The basic idea is to simulate the process that a random walker wanders into the network. The walker starts the journey at random from one of the functions or components that have a risk in history. Then, in each step, the walker may either move at random to a neighboring node or start a new journey with a certain probability.

The R_DSM is a square matrix with diagonal entries representing risk factors and off-diagonal $entries(i, j)$ representing the influencing strength between different risk factors. In the R_DSM, the elements of column represent instigating risk, and the row represent the affected risk. The paper studies the influencing relationship of risks (R_DSM) through functional and component change, and the initiated R_DSM is elicited from the MDM.

Based on the results of the analysis, the paper uses random walk developed by (Gan et al., 2014) to measure the influencing strength between different risk factors. However, the random walk method only studies the relationship between two layers, as the figure 2 shows, the paper calculates the influencing strength of function and component on R_DSM respectively.



	r_1	r_2	r_3	r_4	r_5	r_6
r_1		.59			.42	.16
r_2	.56				.28	.16
r_3				.4		.12
r_4			.76			
r_5	.58	.48				.27
r_6	.32	.32	.21		.27	

(c)

Figure 2. The calculation R_DSM using random walk

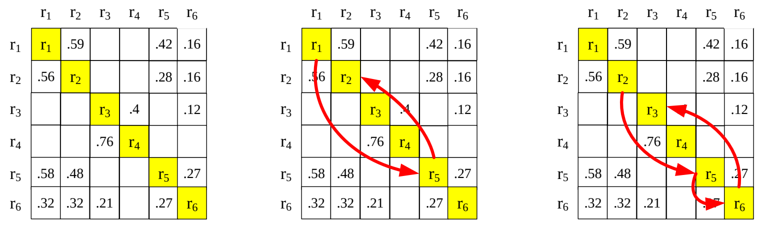
Therefore, based on restart random walk algorithm, the influencing strength resulting from functional change is calculated to be R_DSM_F; then, we can use the similar approach to calculate the value resulting from the component change as R_DSM_C. Hence the integrated influencing strength resulting from the change of function and component can be calculated as formula 1.

$$R\_DSM(i,j) = 1 - [1 - R\_DSM_F(i,j)] \times [1 - R\_DSM_C(i,j)] \qquad (1)$$

4. Analyzing the influencing strength based on direct and indirect risk propagation

4.1 The direct and indirect propagation

The traditional risk analysis mainly focuses on the direct influencing between risk factors. In fact, the influencing relationship between different risk factors is not only directly related but also indirectly influenced by many possible and potential paths. As shown in figure3, the influencing strength from r_1 to r_2 including the direct influencing strength (0.56) and the indirect influencing strength from r_1 to r_2 through intermediate risk r_5 (0.58×0.28).



(a) The direct influencing strength

(b) The influencing strength through an intermediate risk

(c) The influencing strength through two intermediate risks

Figure 3. The example of direct and indirect influencing strength

4.2 The combined influencing strength DSM

The combined influencing strength is defined as the integrated influencing strength of all possible change propagation paths. We assume that the changes would not transmit appreciably beyond three steps, which is a reasonable assumption based on previous research (Koh et al., 2012; Giffin et al., 2009; Clarkson et al., 2004). Therefore, analyzing the combined influencing strength through direct, one and two intermediate risk factors.

(1) The direct influencing strength

As shown in figure3 (a), the $R\_DSM(i,j)$ represents the direct influencing strength risk j on risk i, so the direct influencing strength can be calculated as formula 2.

$$RS^1(i,j) = R\_DSM(i,j) \qquad (2)$$

(2) The influencing strength through an intermediate risk

As shown in figure3 (b), the indirect influencing strength of risk j on risk i through an intermediate risk p $RS^2(i,j)$ can be calculated as formula 3.

$$RS^2(i,j) = 1 - \prod_{p=1}^{N_c} (1 - RS_p^2(i,j)) = 1 - \prod_{p=1}^{N_c} [1 - \mathrm{DSM}(p,j) \times \mathrm{DSM}(i,\mathrm{p})] \qquad (3)$$

Where $i \neq j, i,\ j \neq p$, p is the intermediate risk from j to i and N_c is the number of all conventional risk factors on the path from j to i.

(3)The influencing strength through two intermediate risks

As shown in figure3 (c), the influencing strength of risk j on risk i through two common risk factors, p and q, $RS^3(i,j)$ can be calculated as formula 4.

$$RS^3(i,j) = 1 - \prod_{p,q=1}^{N_c} (1 - RS_{p,q}^3(i,j)) = 1 - \prod_{p,q=1}^{N_c} [1 - DSM(p,j) \times DSM(q,p) \times DSM(i,q)] \quad (4)$$

Where, $i \neq j; i, j \neq p, q$, p and q are the intermediate risks from j to i , N_c is the number of all conventional risk factors on the path from j to i .

So, the combined influencing strength risk j on risk i $CRS(i, j)$ can be calculated as formula 5. $CRS(i, j)$ is defined as the integrated influencing strength of all possible paths.

$$CRS(i,j) = RS^1(i,j) \cup RS^2(i,j) \cup RS^3(i,j) = 1 - \prod_{z=1}^{3}(1 - RS^z(i,j)) \qquad (5)$$

5. An illustrative example

The following case study will illustrate how the model and methodology developed in the preceding sections can be applied in a real-work setting. Based on the research and development of smart-phones, the paper investigates the change of function and component in the project, identify risk factors and determine the potential relationship between different risk factors. Analyzing the change of function and component in the smart-phones projects, the paper builds the E-MDM of "function-component-risk" as shown in figure 4(a).

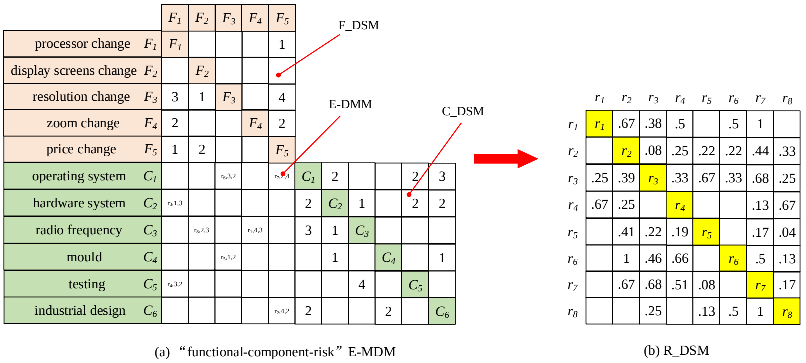


Figure 4. The calculation of initial R_DSM of smart-phones R&D

Based on the initial influencing relationship using MDM, the random walk gives the quantification of influencing strength between risk factor, as shown in figure 5(a). The paper analyzes the combined influencing strength affected by direct and indirect propagation through one and two common risk factors, shown in figure 5 (b) and (c).

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
r_1	r_1	.50	.46	.12	.15	.15	.21	.16
r_2	.29	r_2	.41	.36	.18	.17	.28	.02
r_3	.18	.41	r_3	.53	.18	.18	.16	.03
r_4	.11	.13	.29	r_4	.09	.14	.18	.03
r_5	.07	.27	.26	.18	r_5	.21	.06	.01
r_6	.11	.28	.29	.22	.23	r_6	.22	.07
r_7	.30	.33	.46	.40	.14	.28	r_7	.07
r_8	.24	.11	.08	.02	.06	.08	.10	r_8

(a) The direct influencing strength

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
r_1	r_1	.66	.66	.53	.35	.37	.42	.21
r_2	.44	r_2	.64	.60	.36	.39	.44	.12
r_3	.38	.58	r_3	.66	.35	.37	.39	.10
r_4	.25	.37	.45	r_4	.22	.27	.29	.08
r_5	.23	.44	.45	.41	r_5	.32	.25	.06
r_6	.31	.50	.53	.49	.35	r_6	.37	.12
r_7	.47	.61	.68	.65	.37	.46	r_7	.16
r_8	.31	.30	.29	.18	.16	.18	.21	r_8

(b) The influencing strength through an intermediate risk

	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
r_1	r_1	.72	.71	.55	.39	.41	.46	.22
r_2	.53	r_2	.71	.64	.40	.43	.49	.13
r_3	.44	.67	r_3	.69	.38	.41	.44	.12
r_4	.27	.39	.51	r_4	.26	.29	.33	.10
r_5	.24	.47	.48	.44	r_5	.34	.27	.08
r_6	.33	.53	.56	.51	.38	r_6	.40	.14
r_7	.53	.66	.73	.67	.39	.49	r_7	.19
r_8	.31	.30	.29	.18	.16	.18	.21	r_8

(c) The influencing strength through two intermediate risks

Figure 5. The results of analyzing the influencing strength in the smart-phone projects

From the calculation results can be seen, the combined influencing strength DSM fluctuated whether indirect propagation is considered, such as, the direct influencing strength of r_1 on r_2 is 0.29, and the value is 0.44 through an intermediate risk, and the value is 0.53 through two intermediate risks. The numerical results indicate whether indirect propagation is taken into account when analyzing influencing strength between risk factors has a significant impact on the influencing strength. Therefore, in measuring the influencing strength between different risk factors, the indirect relationship is very important, because the true and combined influencing strength would surely be affected by intermediate risks.

6. Conclusion

To assist managers in facing risks caused by the change of function and components, the paper analyzes the risk factors using the E-MDM of “function-component-risk”. On the basis of identifying risk factors and determining the potential relationship deriving from the change of function and component, we use random walk algorithm to analyze influencing strength. Moreover, the paper analyzes the influencing strength based on direct and indirect risk propagation and calculates the combined influencing strength between different risk factors through all possible paths. The validity of the model and algorithm is verified by a research on development of smart-phones.

Nevertheless, the approach has also some limitations that are outlined in the following. Since this is a mathematical deductive approach, we had to make a few assumptions. For instance, we calculate the R_DSM based on the E-MDM of "function-component-risk" and analyze the risk factors deriving from the change of function and component. In reality, there may also be many other changing factors that may lead to risk, such as, design changes and environmental factors. For further analysis and evaluation, the more detailed and practical changing factors that may lead to risk should be concerned. Moreover, random walk enriches the theory and application of risk, is an interesting issue worth to be studied further.

Acknowledgments

This study was supported by the National Science Foundation of China (No. 71472013, No. 71528005 and No. 71872011)

References

Ackermann, F., Howick, S., Quigley, J., Walls, L., Houghton, T. ,2014. Systemic risk elicitation: using causal maps to engage stakeholders and build a comprehensive view of risks. European Journal of Operational Research, 238(1), 290-299.

Clarkson, P. J., Simons, C., Eckert, C., 2004. Predicting change propagation in complex design. Journal of Mechanical Design, 136(8), 52-68.

Eckert, C., Clarkson, P. J., Zanker, W. ,2004. Change and customization in complex engineering domains. Research in Engineering Design, 15(1), 1-21.

Gan, M., 2014. Walking on a user similarity network towards personalized recommendations. Plos One, 9(12), e114662.

Giffin, M., Weck, O. D., Bounova, G., Keller, R., Eckert, C., Clarkson, P. J., 2009. Change propagation analysis in complex technical systems. Journal of Mechanical Design, 131(8), 183-192.

Koh, E. C. Y., Caldwell, N. H. M., Clarkson, P. J., 2012. A method to assess the effects of engineering change propagation. Research in Engineering Design, 23(4), 329-351.

Qazi, A., Quigley, J., Dickson, A., Kirytopoulos, K., 2016. Project complexity and risk management (ProCRiM): towards modelling project complexity driven risk paths in construction projects. International Journal of Project Management, 34(7), 1183-1198.

Shenhar, A. J., 2001. One size does not fit all projects: exploring classical contingency domains. Management Science, 47(3), 394-414.

Yang qing,Shan chen,Tang er-ling., 2015. Optimization of design structure matri in complex R&D projects based on rework risk propagation and pretreatment.Systems Engineering-Theory & Practice, 35(6), 1501-1508.

Eppinger S, Browning T.,2012. Design Structure Matrix Methods and Applications. MIT Press, Cambridge.

Yang, Q., Kherbachi, S., Hong, Y. S., Shan, C. ,2015. Identifying and managing coordination complexity in global product development project.International Journal of Project Management, 33(7), 1464-1475.

Contact: Qing Yang, University of Science & Technology Beijing, School of Economics and Management, No.30 Xueyuan Road, Haidian District, 100083 Beijing, China, 13911463353, yangqing@manage.ustb.edu.cn.

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Conceptual Framework for Collaborative Risk Management during the Design Phase of Green Building Projects using DSM

Lungie Maseko[1], David Root[2], Venkatachalam Senthilkumar[3]

[1]Student, University of Witwatersrand
[2]Professor, University of Witwatersrand
[3]Assistant Professor, University of Sharjah

Abstract: Construction projects are ambitious in terms of the complexity in its components, structures, design requirements, information flows, stakeholder integration and technological integration particularly in green building projects. As a consequence, management of these projects becomes increasingly integrated; however, risk management has taken little account of these interdisciplinary and iterative trends. This leads to poor risk management outcomes, where traditional risk management practices that rely on allocating risks to specific individual entities are not able to accommodate the collaborative facets. Experienced practitioners were interviewed regarding their current practices and techniques towards managing interdependent design tasks that resulted in inseparable collective risks. Prospective utilization of Dependency Structure Matrices (DSM) and its analysis of identifying the existence of these collaborative Design Risks among the clusters of designs are proposed as a solution in this paper. Since the paper is explorative in terms of the application of the DSM method on identifying and managing the collaborative risk management in green building design, conceptual frameworks are only proposed.

Keywords: Collaboration, Design tasks, Green buildings, Risk sharing

1 Introduction

Building design involves complex and comprehensive work that requires the cooperation of various specialties as collaborating stakeholders (Liu et al., 2014). With the multifaceted nature of projects, building design becomes increasingly difficult and complex. Thus the major shift towards collaborative design approaches (El-Diraby et al., 2017). However the traditionally used planning methods such as CPM and PERT cannot model the iterative nature of design processes (Senthilkumar et al., 2010). Dependency Structure Matrix (DSM) is an effective method developed to model iterative process (Senthilkumar and Varghese, 2013). This study intends to discover how project stakeholders in collaborative teams manage inseparable risks within their different design tasks on green buildings and how DSM can be proved to be effective in representing the design process and managing risk within the design domains.

Collaborative design, demands the process of coordination and cooperation of different stakeholders who share their knowledge in both the design process and the design content (Kleinsmann, 2006), as a means of attaining the unified design goals in the most efficient and effective ways (Liu et al., 2014). Traditionally, risk management has given little consideration to the collaboration within the interdisciplinary and iterative design process. Risk management practices continue to rely on allocating risks to specific individual entities, which is increasingly problematic given the non-coherence of the growing collaborative green building sector, where the design philosophy is holistic and treats the building as a complex integrated system (El-Diraby et al., 2017), that is best designed, and efficiently executed through collaborative practices.

Chiu (2002) defines collaboration "as an activity that requires participation of individuals for sharing information and organizing design tasks and resources." This means that the stakeholders would provide each other with new insights that would enable each participant to fulfill his or her own task without compromising/ affecting the design of others whilst meeting the common objectives of green building. These objectives are typically; to lower energy consumption, lower investment costs, and reduced harmful impacts on the environment and on people (EPBD, 2015). In collaborative designs, tasks are interdependent and iterative (Al Hattab and Hamzeh, 2015). Iteration assists in the progressive generation of knowledge, enabling a degree of concurrency and integrating necessary changes, although it can also increase the duration and cost of a project (Wynn and Eckert, 2017). Managing where and how iteration occurs is thus an important issue in practice in order to mitigate these additional costs due to non-value adding iterations or rework. This can be a challenge where it relates to risk. Consequently, the need for stakeholder collaboration and risk management to provide an effective way of managing risks is, present and unavoidable. Risks are inherent in all complex projects (Peckiene et al., 2013) and how risks are shared among stakeholders in the design phase is mostly governed by the dynamic evolution of management. Hence, any dynamic approach needs effective risk management and collaborative efforts among project stakeholders (Lam et al., 2007; Gomes et al., 2016).

Every Collaborative Risk Management (CRM) solution is impacted by people, technology and the nature of multidisciplinary tasks and participants who need to deliver a holistic risk system with a final design product. CRM is about the dynamic management of risk (Rahman and Kumaraswamy, 2005) which plays a major role in achieving value-for-money and cost-efficiency in designing complex projects. Typically, for design only, an activity-based DSM methodology would be used for dependencies and interface identification (Senthilkumar et al., 2010). Yet, inseparable design risks need to be resolved in a holistic manner in all aspects of the green building design process, hence the need to explore alternate methods to formulate the DSM.

2 Green Buildings Management with Collaborative Risk Management Principles

Green buildings (GB) are structures designed to promote efficient use of resources (e.g., energy, water, and materials) and that promotes sustainability (WCED, 1987). The US Environmental Protection Agency (2016) defines green building as: "the practice of creating structures and using processes that are environmentally responsible and resource-efficient throughout a building's life-cycle from siting to design, construction, operation, maintenance, renovation and deconstruction."

Green Building designs are complex undertakings that have given rise to reciprocal interdependencies between multiple and diverse stakeholders, hence the high dependence on information, followed by the connectedness of tasks (Austin et al., 2002; Ahn et al., 2016). Further, Bakhshi et al. (2016) defines GBs complexity as an intricate arrangement of varied interrelated parts in which the elements can change and evolve constantly with an effect on project objectives. Yet, they are the most effective solutions to increase the efficiency of buildings through resource utilization and recycling, mitigating the negative impact of the construction industry on the environment (Zuo and Zhao 2014). This has been made possible through *inter alia,* mutual collaboration, adjustments towards working collectively and responding to emergent, unforeseen problems in real-time. However project realities are such that current risk practices promote competitive attitudes between the project stakeholders involved because they tend to work for their self-interests and thus safe-guard their existence in the project (Alsalman 2012). So, it is vital to change, not only risk management (RM) practices; but, mindsets to shift towards mutual adjustments and rapid adaptation where stakeholders will be in a give-and-take interdependence (Morris 2013). The change from traditional RM to CRM is loaded with uncertainties on risk sharing among all project stakeholders and their response to this cultural shift.

Risk sharing requires all stakeholders within complex projects to take a closer look at their own risk universes. It is a useful method for handling complex designs (Melese et al., 2016), and a collaborative way of managing risks by taking advantage of the different views from different stakeholders (Olander, 2007). CRM appears to be a relevant problem as it emphasizes equitable and balanced risk sharing among contracting stakeholders who want to eliminate improper or unfavorable risk sharing outcomes which result in cost and time overrun and, undoubtedly, in legal disputes (Loosemore and McCarthy, 2008).

In this vein, the traditional tools (PERT, Gantt and CPM) based on workflows have failed to address interdependency (feedback and iteration) and would not be suitable for modeling information flows that determine the design phase (Yassine et al., 1999). Hence, DSM is identified as a useful tool for coping with design issues (Steward, 1981). The DSM matrix can be used to identify appropriate stakeholders, teams, and the ideal sequence of the tasks (Lindemann, 2009). A DSM involves a square matrix with an equal number of rows and columns that shows relationships between tasks in a system (Eppinger and Browning, 2012). Collectively, these complexities and interdependencies of tasks result in inseparable design risks. These kind of risks cannot be transferred or allocated to an individual, but would have to be shared collaboratively. How then do project stakeholders in collaborative teams deal with inseparable risks within their different design tasks?

3 Identifying Inseparable Risks within the Design Phase

The emphasis of effective RM in dealing with the broad spectrum of risks is to move beyond the traditional RM mechanics to examine the sources of unknown risks (Jarkas and Haupt 2015). Though the construction industry has long managed to identify and analyse known risks, it has recognized that dealing with the hidden, less obvious aspects of uncertainty is complicated and results in inseparable risks, and this requires practitioners to be more proactive in their approach (Smith and Merritt 2002). Inseparable risks arise from uncertainties, ambiguities and arrays of risk factors that are intricately connected (Thamhain, 2013).

In practice, a typical approach to risks is trying to identify them as early as possible and respond to them as quickly as possible once identified (Kim, 2017). However, green projects anticipate unidentified risks, also known as ‘unknown unknowns’ that have traditionally been underemphasized by risk management (Thamhain, 2013). It is difficult to trace the causes and culprits of these unknown unknowns as they require inventive risk handling decisions on risk allocation (Jin et al., 2017). Predicting and controlling such unknown risks has also developed impractical risk preferences for some project stakeholders because they sometimes actively ignore those (Alles 2009). These risk attitudes have made the risk sharing process challenging (Walker, 2015).

The goal of identifying inseparable risks is to make the process of risk sharing more efficient through planning and coordination by mutual adjustment, so as to get a better information flow in design (Fundli and Drevland 2014). Design risks have been classified in a number of ways. Arguing that risks arise as a result of interactions between stakeholders, technological interoperability and organizational factors, Smith et al. (2009) suggested that they may be grouped as either involuntary or voluntary, depending on whether the incidents that create the risk are uncertain or beyond the control of the people in charge.

The increasing complexity of projects and knowledge processes, makes it imperative for stakeholders to be keenly aware of the intricate connections of risk variables among complex systems and processes (Thamain 2013), this limits the effectiveness of traditional RM methods. Stakeholders argue that no single person has all the smarts and insight for assessing multi-variable risks and their cascading effects (Hartono et al., 2014). Project stakeholders realize that, while there may be good RM methods which provide a critically important toolset for risk management, it takes the collective thinking and collaboration of all the stakeholders to identify and deal with the complexity of inseparable risks in green building projects.

4 Research method and Data analysis

A case study strategy was adopted in this research, as case studies typically use a variety of data collection methods such as interviews, questionnaires, and observations (Eisenhardt, 1989). CRM is a relatively innovative concept in South Africa and, therefore, it is important to obtain a detailed and comprehensive view of it by investigating it in past and ongoing projects. In particular, how CRM is managed in design processes and how various stakeholders manage inseparable risks, were areas of interest.

The case study data to this investigation was collected through semi-structured interviews; with a mixture of open and close-ended questions (Brink, 2014), where participants were asked - stakeholder techniques on carrying out inseparable tasks, as well as their options and suggestions on CRM processes of green projects.

The case studies comprise of a ‘completed project’ and a ‘project in its design phase’. The completed project is of residential apartments in the V&A Waterfront in Cape Town, South Africa and the project team of this case study reflects on the problems they faced.

The other case study is an academic Forensic Pathology Facility in Johannesburg, South Africa; this project is in its design phase and the project team is still engaging with their risks. In both projects, numbers of stakeholders with varying backgrounds were involved and it thus was interesting to see how CRM could be applied. The objective for the interviews was to explore the possible challenges that had not been identified in the literature review of managing green construction projects; and identifying areas where inseparable risks were and could be managed.

The analysis and interpretation of research data form the major part of the research (Amaratunga et al., 2002). The methodical process used was the DSM, which is a square matrix that focuses on dependencies between elements of one domain like people-people, component –component and task-task sequence relationships. Then, the Domain Mapping Matrix (DMM) was used as it examines the interactions across domains to represent enriched analysis results that provide an expanded view of the complex system (Bartolomei et al., 2007). When applied, a DMM was constructed to map out the interdependencies, interactions, and exchange of information from design tasks and risks, identifying the optimal sequence of tasks, risk interactions and iterations across domains (Yang et al., 2014). The combination of square DSM and rectangular DMM is called Multiple Domain Matrix (MDM) where useful information is provided using intra- and inter-domain networks (Lindemann and Maurer, 2007).

Also, the DSM process was utilized to identify clusters (Browning, 2015) in a matrix analysis approach that minimizes iterations and enhances efficiency in risk management (Jaber et al., 2015). The high interaction of clusters encouraged stakeholders to collaborate, communicate and coordinate better, so to identify and examine interfaces between the clusters and keep iterations at a minimum; minimizing the number of task dependencies (Austin et al., 2001).

5 Findings

This research is still on going and, more interviews are still to be conducted. For now, 15 semi-structured interviews with different experts were conducted to understand the current risk allocation practices and the way inseparable risks can be managed in collaborative circumstances.

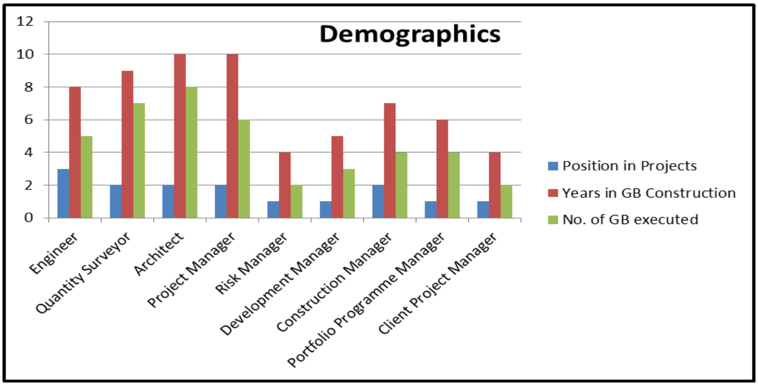


Figure 1.Profile of Respondents

Based on the interviews **Figure 1** represents the research demographics and 65% of the projects done by the respondents were in Gauteng, 15% in the Western Cape and the other 15% was in different South African provinces, with only 5% on international projects.

Data analysis focused on how each project managed its CRM practices. An analysis on sources of design risk, project risk management process, collaborative activities and the design process results was achieved. Second, cross-case analysis was performed in order to examine similarities and differences in the projects. Based on the categories presented by Burns and Stalker (1961) and Geraldi (2008), comparisons on how different risk management systems were used in the two projects affected CRM.

Figure 2 shows how the use of DSM/DMM aims to handle Collaborative Risks (CR) during the design phase by identifying interdependencies. On DMM people-activities and people-components, communication plans on how identified CR will be managed should be discussed by stakeholders. And, these matrices will potentially identify clusters of risks; improve coordination and management, for CR to be shared equitably.

	Stakeholders	**Design Tasks**	**Design Components**
Stakeholders	People DSM	People Activity Domain Mapping Matrix	People Building Components Domain Mapping Matrix
Design Tasks		Activity DSM	Activity Building Component Domain Mapping Matrix
Design Components			Component DSM

Figure 2.MDM Mapping System for capturing the Design Process Interfaces in various domains

For collaborative activities as shown in **Figure 3**, interviews were analyzed using the DMM matrix to plot the information and map-out interdependencies between the stakeholder and the RM activities, it is a Domain Mapping Matrix which captures the interrelationships among various stakeholders on specific RM tasks. The purpose of this matrix is to illustrate the interactions capturing procedure which can be adopted using DMM and their useful contribution in this process of uncertainty reduction and management.

Interdependencies of varying strengths are identified across activities and by clustering; this DMM identifies areas between tasks and stakeholders that require a high level of coordination and integration. Interfaces between these activities indicate the people who must communicate to transfer information. But, inseparable risks still need to be allocated on design processes. A fair and equitable risk sharing is essential to ensuring a successful delivery of a project design. Stakeholders must work collaboratively to seek an equitable sharing of risk based on an appropriate methodology that seeks to allocate design risks in an efficient manner and with specific considerations. In doing so, the intention will be to reduce project disputes and benefit of all parties.

Collaborative Activities	Owner	Architect	Quantity Surveyor	Structural designer	Electrical designer	Mechanical and plumbing designer	Fire protection designer	Main contractor	Sub-contractors (Mechanical, Plumbing, Electrical, Fire)	Process and technology manager
Determine reciprocal responsibility of stakeholders	x	x	x	x	x	x	x	x	x	x
Determine how to allocate benefits and risks	x	x	x	x	x	x	x	x	x	x
Establish mechanism of conflict coordination	x	x	x	x	x	x	x	x	x	x
Analyze functional requirements of the design		x	x	x	x	x	x	x	x	
Determine design criteria for each specialty		x	x	x	x	x	x	x		
Determine standards for exchanging BIM data	x	x	x	x	x	x	x	x		x
Determine time control points for design tasks	x	x	x	x	x	x	x	x	x	x
Examine the schedule jointly	x	x	x	x	x	x	x	x	x	x
Examine the site design jointly	x	x	x	x	x	x	x	x	x	
Considerations of environmentally safe methods of construction		x	x	x			x	x		x

Figure 3.Collaborative Activities

Figure 4 is a conceptual framework of the application of DSM methods. Participative use of DSM/DMM/MDM methods will create situations for stakeholders to discuss their tasks, information needed, risks anticipated and the interdependencies. This will enable them to outline the design of the information exchange process, engaging all involved. These methods are enabling tools to create crucial communication lines, to reduce assumptions and uncertainty between stakeholders. The combinations of these matrices provide improved decision support for stakeholders on the purposes in the conceptual framework; clustering analysis being the decisive factor to create understanding on the collaborative risk context; accountability and transparency will then be achieved and risk sharing will be done fairly.

Discussion

Due to the dynamic, complex nature of green designs and the interplay of multi-stakeholders, RM processes used require collaboration between the stakeholders. The collaboration needed has been amplified by the interdependencies of stakeholders and their dependable tasks which resulted to inseparable risks. The use of DSM/DMM/MDM methods to improve RM practices is a solution towards equitable and balanced risk sharing.

	DSM/DMM/MDM	Purpose	Analysis	Decision Rule/ Application
1.	MDM among People, Activity, Component	To capture the big picture on the overall Project CRM	Clustering	To identify and strategies the project procurement method
2.	DSM People	To Allocate CR to appropriate people who shares the dependent activities/ components	Clustering	Clustering to shape the project communication management protocol
3	DSM Activities	To identify and sequence the design process activities which are collaborative and iterative in nature	Partitioning	Partitioning and Sequencing to avoid schedule delay risk
4	DSM Components	To come up with the work packages, the components which are highly interactive should be procured as a single work package	Clustering	Clustering to come up with work packages and sequencing of work to avoid risk due to lack of collaboration
5	DMM People Activity	To identify collaborative risks within interdependent activities and share the risks to appropriate people	Clustering	To identify and allocate the responsibility of CR, allocate and manage resources
6	DMM People Components	Assign single work packages to suitable people who will share CR equitably	Clustering	To identify and allocate the responsibility of CR, and check performance related risks
7	DMM Activity Component	Sequence design process activities with fitting work package components and comprehend interrelationships	Clustering	To identify and allocate Risk management provisions on design processes against the components.

Figure 4.Conceptual framework on application of DSM methods on managing the CR in Green Building Design Process

The application of the DSM/DMM/MDM methodologies is still limited in CRM practices of designing GB. Future work is required to determine the procedure to manage equitable risk sharing using these methodologies efficiently as means to improve the stakeholders' behavior in their interactions across multi-domains through work package allocation, communication and collaboration clauses on the contracts, partnership and alliancing arrangement, penalty clauses, contingency allocation, etc. Though the study can also be expecting some limitations, the proposed conceptual method shows potential for improvement on the collaborative risk management during the design of green building projects. Further validation is needed to claim the rigor of this finding, which will be provided in the forthcoming publications.

References

Ahn, Y. H., Jung, C. W., Suh, M. and Jeon, M. H., 2016. Integrated Construction Process for Green Building, Procedia Engineering 145: 670-676.

Al Hattab, M. and Hamzeh, F., 2015. Using social network theory and simulation to compare traditional versus BIM–lean practice for design error management, Automation in Construction 52: 59-69 .

Alsalman, A. A. 2012. Construction risks allocation: Optimal risk allocation decision support model (Doctoral dissertation), Oregon State University.

Austin, S., Baldwin, A., Hammond, J., Murray, M., Root, D., Thomson, D. and Thorpe, A., 2001. Design Chains, A Handbook for Integrated Collaborative Design, Thomas Telford Publishing, London.

Austin, S., Newton, A., Steele, J. and Waskett, P., 2002. Modelling and managing project complexity, International Journal of Project Management 20(3): 191-198.

Bakhshi, J., Ireland, V. and Gorod, A., 2016. Clarifying the project complexity construct: Past, present and future, International Journal of Project Management, 34(7): 1199-1213

Bartolomei, J., Hastings, D., de Neufville, R. and Rhodes, H., 2007. Engineering Systems Multiple Domain Matrix: An Organizing Framework for Modeling Largescale Complex Systems, Systems Engineering 15(1): 41–61.

Brink, S. 2014. Interview, New York: Springer.

Browning, T. R., 2015. Design Structure Matrix Extensions and Innovations: a survey and new opportunities, IEEE Transactions on Engineering Management.

Chiu, C. H., 2002. The effects of collaborative teamwork on secondary science, Journal of Computer Assisted Learning 18 (3): 262-271.

Eisenhardt, E. M., 1989. Building theories from case study research, Academy of Management Review 14(4): 532-550.

El-Diraby, T., Krijnen,T. and Papagelis, M., 2017. BIM-based collaborative design and socio-technical analytics of green buildings, Automation in Construction 82: 59-74.

El-Sayegh, S. M., 2008 Risk assessment and allocation in the UAE construction industry, International Journal of Project Management 26(4): 431-438.

Eppinger, S. D. and Browning, T. R., 2012. Design structure matrix methods and applications, MIT press.

EPA – 2016. US Environmental Protection Agency Definition of Green Building https://archive.epa.gov/greenbuilding/web/html/about.html

EPBD – 2015. Energy Performance of Building Directive, compliance study, Written by ICF International, December. https://ec.europa.eu/energy/sites/ener/files/documents

Geraldi, J. G. (2008) The balance between order and chaos in multi-project firms: A conceptual model, International Journal of Project Management, 26 (4): 348-356.

Hartono, B., Sulistyo, S. and Hasmoro, D., 2014. Project risk: Theoretical concepts and stakeholders' perspectives, International Journal of Project Management, 32(3): 400-411.

Jarkas, A. M and Haupt, T. C., 2015. Major construction risk factors considered by general contractors in Qatar, Journal of Engineering, Design and Technology 13(1): 165 – 194.

Kim, S. D., 2017. Characterization of unknown unknowns using separation principles in case study on Deepwater Horizon oil spill, Journal of Risk Research, 20 (1): 151–168.

Kleinmann, M. S., 2006. Understanding Collaborative Design (Thesis PhD), Technische Universiteit Delft, Nederlands.
Lam, K. C., Wang, D., Lee, P. T., and Tsang, Y. T., 2007. Modelling risk allocation decision in construction contracts, International Journal of Project Management 25(5): 485-493.
Lindemann, U. and Maurer, M., 2007. Facing Multi-Domain Complexity in Product Development, The Future of Product Development.
Lindemann, U., 2009. The design structure matrix (DSM), Retrieved from http://www.dsmweb.org/en/dsm.html
Liu, W., Guo, H. and Skitmore, M., 2014. A BIM-Based Collaborative Design Platform for Variegated Specialty Design, ICCREM 2014: Smart Construction and Management in the Context of New Technology, American Society of Civil Engineers (ASCE), Kunming, China, 320-329.
Morris, P. W. G., 2013. Reconstructing Project Management, Wiley-Blackwell, Chichester.
Olander, S., 2007 Stakeholder impact analysis in construction project management. Construction Management and Economics 25(3): 277–287.
Peckiene, A., Komarovska, A. and Ustinovicius, L., 2013. Overview of risk allocation between construction parties. Procedia Engineering 57: 889-894.
Rahman, M. M. and Kumaraswamy, M. M., 2005. Assembling integrated project teams for joint risk management, Construction Management and Economics 23(4): 365-375.
Runeson, P., Host, M., Rainer, A. and Regnell, B., 2012. Case study research in software engineering: Guidelines and examples, John Wiley and Sons, Inc. Hoboken: New Jersey.
Saunders, M., Lewis, P. and Thornhill, A., 2016. Research methods for Business students, 7th Ed, Italy: Pearson Education Limited.
Schöttle, A., Haghsheno, S., and Gehbauer, F., 2014. Defining Cooperation and Collaboration in the Context of Lean Construction, Proceedings of the 22th Conference of the International Group for Lean Construction, Oslo, Norway.
Senthilkumar, V., Varghese, K. and Chandran, A., 2010. A web-based system for design interface management of construction projects, Automation in Construction, 19(2):197-21
Senthilkumar, V. and Varghese, K., 2013. Case Study–Based Testing of Design Interface Management System, Journal of Management in Engineering, 29(3): 279-288
Smith, N. J., Mernam T. and Jobling, P., 2009. Managing risk: In construction projects, 2nd Edn, Malden, MA: Blackwell Science Inc.
Steward, D. V., 1981. Design structure system: A method for managing the design of complex systems, IEEE transactions on Engineering Management, 28(3): 71-74.
Thamhain, H., 2013. Managing risks in complex projects, Project Management Journal 44(2): 20
WCED – 1987. World Commission on Environment and Development. Our Common Future (Chapter 2): Towards Sustainable Development. http://www.un-documents.net/ocf-02.html
Yang, R. J. and Zou, P. X., 2014. Stakeholder-associated risks and their interactions in complex green building projects: A social network model, Building and Environment, 73: 208-222
Yassine, A. A., Chelst, K. R. and Falkenburg, D. R., 1999. A decision analytic framework for evaluating concurrent engineering, IEEE Transactions on Engineering Management, 46(2): 144-157.
Zuo, J. and Zhao, Z. Y., 2014. Green building research–current status and future agenda: A review, Renewable and Sustainable Energy Reviews, vol.30: 271-281.

Contact: L. Maseko, University of Witwatersrand, Construction Economics and Management, 1 Jan Smuts Avenue, PO Box 20, Wits, South Africa, +27716290992, lungie.maseko@wits.ac.za

Part II: Complex Organizations

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Simulation-based Value Analysis of Organizational Complexity in Product Development Projects

Nikolaus von Bary[1], Eric Rebentisch[2], Lucia Becerril[1]

[1]Technical University of Munich
[2]Massachusetts Institute of Technology

Abstract: This paper analyzes the relation between organizational complexity and the value of product development projects. For this end, an agent-based simulation that incorporates essential factors of complexity is used. During each simulation run, complexity is measured, and the project outcome is captured. Additionally, other behavioral measurements are collected to show the behavior on the agent level. Throughout different scenarios, the simulation differentiates between low and high task difficulty, segregated and integrated communication patterns as well as strict, chaotic, and decentralized organizations. The results of the analysis show, that higher complexity than the possible minimum is advantageous. For every simulated scenario, the optimal values of complexity are different. Overall, a certain amount of organizational complexity appears to be favorable for the project setup. Yet, exceeding this value is harmful to a project, which is consistent with the broad negative opinion towards complexity in literature.

Keywords: complexity, organization, organizational behavior, product development projects, agent-based simulation

1 Introduction

Since complexity affects all areas of product development (Götzfried, 2013), it continuously gains more relevance in research. Research areas include e.g. product complexity (Sinha, 2014), process complexity (Browning & Eppinger, 2002), organizational complexity (Rebentisch et al., 2016) or sourcing complexity (Novak & Eppinger, 2001). Yet, the widespread opinion among current research across different areas is similar: Complexity leads to a rise in cost, development time, and risk of project failure, and therefore needs to be reduced, avoided, or managed (Birkinshaw & Heywood, 2010; Carlucci, Lerro, & Skaržauskienė, 2010; Danilovic & Browning, 2007; Götzfried, 2013; Lissack & Gunz, 2005; Oehmen, Thuesen, Ruiz, & Geraldi, 2015; Qureshi & Kang, 2015). This paper recognizes certain harmful aspects of complexity. However, the overall negative stance towards complexity is questioned. The goal of this study is to challenge the purely negative view on complexity. Therefore, the following two working hypotheses are formulated:

H1: Complexity in project organization is not necessarily disadvantageous and therefore does not have to be reduced in every case.

H2: For certain project organization characteristics, a higher level of complexity other than the lowest possible level can lead to reduced project time and cost.

To validate the hypotheses, we examine organizational complexity through an agent-based simulation. An agent-based simulation is chosen for several reasons. First, by using a simulative approach, changes can be easily tested in the organization as a real organization would not allow. Additionally, due to its bottom-up character, emergent behavior can be observed in an agent-based simulation which is critical for complexity analysis. The simulation is not created by looking at the top-level system behavior but by modeling the members of an organization. Thus, new insights for the overall behavior can be gained that were not specifically modeled.

2 Theoretical background

This section summarizes findings and concepts of research regarding product development projects that are relevant to the created model. Then, the theory of organizational complexity is introduced, which forms the basis for implementation of complexity in the simulation and its measurement.

2.1 Product development projects

A product development project is a collection of interconnected activities by a certain number of people over a period of time with the goal to achieve parameters and characteristics of a new product (Clark, 1989). Turner (2008) calls a project an organization within an organization that has the ability to use assigned resources. Thereby, he references to the structure of a project which is very similar to an organizational structure. The similarity of structures stems from the project setup. The project structure is a combination of the organization and the pursued product (Eppinger, 2002; Sinha, 2014). The setup of the project has a strong influence on project success. There is a high linkage between organizational structure and the resulting product structure (Eppinger, 2009; Eppinger & Browning, 2012; Luna & Eppinger, 2015). Luna and Eppinger show a very high predictability of interaction between persons or teams by analyzing product structure. Morelli, Eppinger, and Gulati (1995) claim a predictability of interactions up to 80 %. Since the organizational structure is created first, the organization influences the product and not the other way around (Sosa, Eppinger, & Rowles, 2004).

In addition to the project setup, for a structured and effective project course, a defined product development process is needed. A process is defined as the sequence of certain activities with the use of information, knowledge and material resources (Lindemann, 2006; Ponn & Lindemann, 2011). Processes are made up of individual tasks with three different kinds of dependences between them: dependent, independent and interdependent (Eppinger, Whitney, Smith, & Gebala, 1994). Interdependent tasks have the most impact on complexity (see section 2.2). Thus, in the simulation of this paper, all tasks that are modeled are interdependent tasks.

A project always has one specific desired outcome, which is the main goal of the project (Clark, 1989; Turner, 2008, p. 2). Yet, reaching that goal can be achieved in multiple ways providing an additional set of parameters on project performance: cost, quality and time. These three performance goals are commonly visualized in the triangle of goals (Atkinson, 1999; Kerzner, 2013, p. 31). In his paper, quality is used as a defined model outcome. Cost and time are used to measure project performance.

2.2 Organizational complexity

The term "complexity" does not have an exact definition despite its widespread use (Schwandt, 2009; Weber, 2005). Sturtevant (2013) tries to illustrate complexity using the distinction between complicated and complex. He describes complicated systems as too big or too detailed for one single person to fully understand (Sturtevant, 2013). On the other hand, complex systems are characterized by strange behavior due to unanticipated interactions between elements (Sturtevant, 2013). The missing definition and the knowledge of influencing factors makes it possible to model complexity by including the influencing factors of complexity in the model.

Patzak (1982, p. 22f) focuses on the two factors variety and connectivity inside a system. In his definition, connectivity is composed of the types of relations and the number of relations. Variety, on the other hand, is described by the types and number of elements that are involved in a system. A more extensive look into complexity factors is provided by Steger, Amann, and Maznevski (2007, p. 4f). They describe complexity as a result of four different factors: diversity, interdependence, ambiguity and flux (Steger et al., 2007, p. 4f). Diversity is the plurality of number and types of elements which consists of the multiplicity and the variety of elements (Patzak, 1982, p. 22; Schwandt, 2009). It describes internal and external factors of a system. Therefore, even a simple system can experience complexity due to a diverse environment (Steger et al., 2007, p. 4f). Higher interdependence, generally, leads to increased complexity (Schwandt, 2009). Interdependence within a system is determined by the system structure. For example, modular systems generally have lower interdependence within the system than network structures. Ambiguity describes the uncertainty that comes from an unpredictable system. It is strongly impacted by the availability and clarity of information (Schwandt, 2009). Lastly, flux describes the rate of change of a system and its surroundings (Schwandt, 2009).

When it comes to complexity of systems, emergence is an additional element that needs to be addressed. Lissack and Letiche (2002) call emergence the difference between a system and the combination of its parts. Emergence are behavioral patterns or functions that occur when people or objects are combined together. Those patterns are different than the individual elements acting alone. It is a complex variety of simple rules of individual behavior that ultimately leads to emergence. The problem about dealing with emergence is the difficulty in predicting it (Lissack & Letiche, 2002).

For this paper, the general theory of complexity is adapted for organizational complexity. Dooley (2002) describes organizational complexity only as the variety of different elements in an organization. However, all four of the factors that describe complexity in general can be transferred onto organizational complexity (Rebentisch et al., 2016). They define eight clusters of factors that influence organizational complexity: Interdependence, operating standard procedure, objective or incentive alignment, information systems and tools alignment, location, personality, culture, and management hierarchy. These clusters are described in more detail in Rebentisch et al. (2016).

There are several methods of measurement for organizational complexity (Efatmaneshnik & Ryan, 2015; Sinha & de Weck, 2013; Vidal, Marle, & Bocquet, 2011). This paper uses the method by Rebentisch et al. (2016), which is based on the quantitative measurements of complexity by Sinha and de Weck (2013). Sinha and de Weck (2013) introduce the following formula:

$$C = C_1 + C_2 C_3 \qquad (1)$$

The complexity C is comprised from the component complexity C_1, the complexity of interdependence C_2, and the architectural complexity C_3. Yet, the complexity of single components in an organization would be the complexity of humans. Reliably quantifying the complexity of a human is questionable. Thus, for the measurement of organizational complexity, C_1 is not applicable (Rebentisch et al., 2016). In comparison to a product, an organization has intra-group and inter-group relations. Rebentisch et al. (2016) therefore adjust formula (1) to the following:

$$C = C_{2G} * C_{3G} + C_{2O} * C_{3O} \qquad (2)$$

In this equation, the index *G* represents the group level and the index *O* stands for the organizational level. The indices *2* and *3* still express the complexity of the interactions and the complexity of the organizational architecture.

3 Methodology

This chapter describes the methodology used in this paper. First, the research questions of this work are presented. Then, an introduction in agent-based modeling is given.

3.1 Working hypotheses

The research study behind this paper aims to reveal insights about the influence of project organization complexity on the project value.

To verify the hypotheses posed in chapter 1, following research questions are addressed:

- Is there an optimal level of project organization complexity for maximized project value and which organizational characteristic does it depend on?
- Can certain forms of project complexity be used to systematically improve projects or are all forms harmful to projects?
- How should certain project characteristics be tailored to reach optimal project value?

3.2 Agent-based modeling

In agent-based modeling, the agents are modeled as conscious and independent individuals that are influenced by their environment and make their decisions based on a set of rules (Bonabeau, 2002; Macal & North, 2006, 2010; Rouse & Boff, 2005, p. 323). As agent-based modeling is a study of many individuals working together in a given structure towards a common goal, it is applicable for the study of organizations (Bonabeau, 2002; Epstein, 1999). The interactions between just a few agents with a simple set of rules can already lead to high complexity and unforeseen results (Gilbert & Troitzsch, 2005, p. 10). This is based on the so called emergent behavior (Epstein & Axtell, 1996, p. 33). Emergence is a phenomenon where interactions between objects on a lower level create an object on a higher level of abstraction (Gilbert & Troitzsch, 2005, p. 11).

The structure of agent-based models is typically made out of three elements (Macal & North, 2010): Agents, relationships between agents, and the agents' environment. To model these three elements, the Mesa library is used for this paper. It is a Python-based library that is specifically designed for agent-based modeling (Masad & Kazil, 2015).

4 Design of the agent-based model

For the model in this paper, two types of agents which comprise the development process are modeled: tasks and developers. The tasks are defined in the beginning and are worked on by the developers. The simulation is over when all tasks are completed. Each task and developer has a unique set of attributes which influence its actions. Developers move around in the artificial space and have the choice between working on tasks or communicating. Working on tasks has the chance of advancing the task progress or making a mistake which creates rework. Communication enhances the developers' knowledge. The choice is based on their personality which is made up of the following four characteristics: task allocation preference, tendency for disobedience and decentralized decision making, and knowledge of the project. Task allocation states the preference between working and communicating. Tendency for disobedience defines the probability of a developer to not act as defined by rules in an unproductive way. Decentralized decision-making leads to an action that is not defined, yet productive for the project. Lastly, knowledge of the project is one developer's specific knowledge of the project. The characteristics focus on the developer's behavior. A simulation process is developed which does not include the human thinking but rather the resulting decision-making mechanism.

In the artificial space, the distance between two developers' locations reduces the probability for communication. Additionally, the artificial space is used to model discovery of work. In development projects, work gets overlooked or is not discovered, leading to project delay. Tasks move randomly throughout the artificial space

The pattern of interaction between the developer agents in the simulation is defined by a Dependency and Structure Matrix (DSM). The interaction DSM is based on the DSM that describes the product architecture of the product being developed. As previously described, the product DSM and interaction patterns have very high congruence (Browning & Eppinger, 2002; Eppinger & Browning, 2012; Morelli et al., 1995; Sosa et al., 2004). If elements of a product are connected in a physical way or by any other type of interface, the chance of interaction between the responsible developers increases significantly. As described in section 2.1, Morelli et al. (1995) show predictability of interactions above 80 %. This makes the simplification valid to also use the product structure DSM as the definition of communication patterns in the organization.

The simulation is time based. In each step of the model, the same amount of time passes, and agents' actions are initialized. During each step, data is collected, and measurements are taken. Collected data include time, amount of work, rework, and communication, knowledge, task advancement symmetry, and the distribution of knowledge. The calculations are based on the measurement of organizational complexity from Rebentisch et al. (2016) and adapted to the numeric nature of the simulation. The most important measurement is organizational complexity. For this purpose, measured interdependence between developers is weighted with cultural complexity, interaction complexity, and allocation complexity. This calculation forms a complexity heat map. All entries of the heat map are then summed up and multiplied by the architectural complexity of the communication pattern based on the numeric complexity measurement of Sinha and de

Weck (2013). This leads to a single value measurement of organizational complexity that incorporates the influencing factors of complexity.

Another calculated measurement is the value of a project. As shown previously, project success can be measured with three dimensions: quality, time and cost. In this model, quality is defined to be equal in every model run, as a simulation is finished when all tasks are done with satisfactory quality. The measurement of quality is therefore not applicable. The two measurable dimensions of the project triangle are combined. The project value is defined as a standardized measurement on the scale of zero to one. Both, duration and cost, are weighted equally and comprise half of the value. The increase of both parameters is negative for project value.

5 Results

The main result lies in comparing different projects and how various organizational behaviors perform for those projects. This paper tests four different project types and three different organizational behaviors. The four projects are constructed by combining two factors: task difficulty and the type of interaction-DSM. The different test cases are shown in Figure 1. The strict organization is defined to operate very close to the guidelines. Developers allocate work as they are told. The chaotic organization sets itself apart through high developer free-will and the high resulting uncertainty. Lastly, in the decentralized organization, developers are encouraged to do the work that is smartest from their perspective.

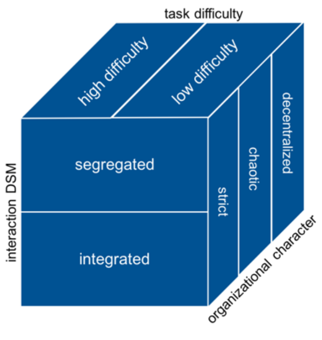


Figure 1. Combination of four different scenarios (interaction DSM and task difficulty) and the three different organizational characters (own illustration)

Analyzing the overall project value over the organizational complexity in Figure 2, it is shown that there is an optimal project value for the simulated projects. On the left side of the figure, the blue lines represent the integrated interaction DSM, whereas the orange lines show the segregated interactions. The lighter shading displays the low difficulty projects and the more saturated lines indicate higher difficulty. Since the outcome of all simulations is assumed to be equal, the more difficult projects have a lower value. They result in the same outcome with higher effort. The segregated scenarios demand more complexity for the optimal project value. This leads to the answer of the first research question. The maximum of the graph suggests a maximum of project value. The different curves show the dependability of that optimal value on a variety of project factors, including project difficulty and interaction patterns.

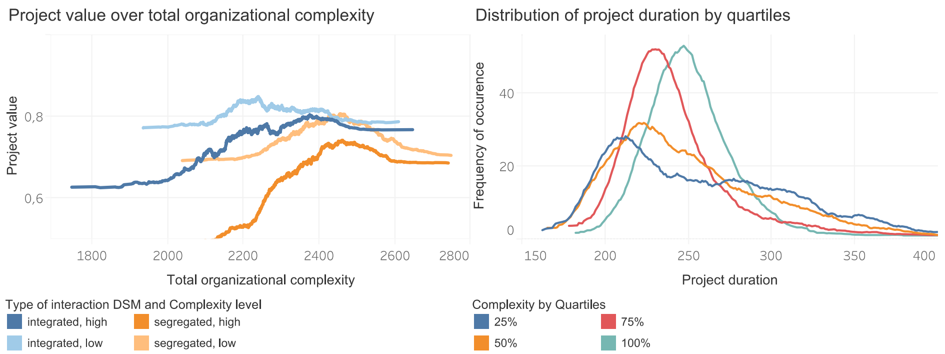


Figure 2. Influence of organizational complexity on project duration

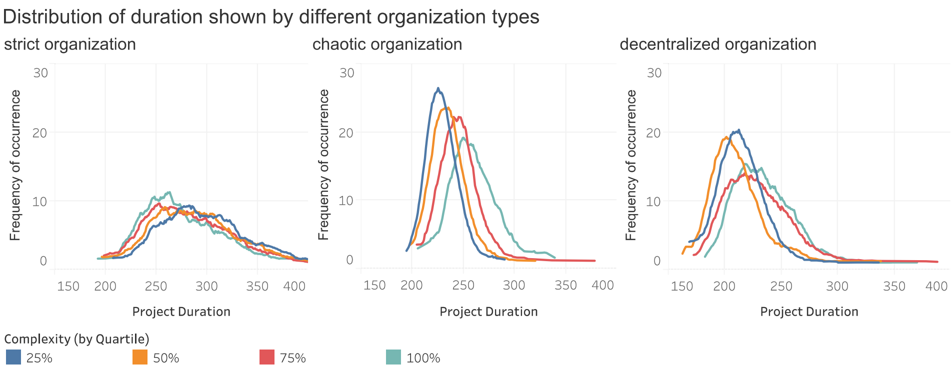


Figure 3. Different levels of complexity are optimal for the various organizational behaviors

On the left side of Figure 2, the frequency of occurrence of different project durations sorted by complexity quartiles is shown. In Figure 3, the three different organizational behaviors are shown in the various graphs. For the strict organization, the highest complexity quartile performs best. For the chaotic organization, the quartile with the lowest complexity achieves the best results. Yet, these two organization types are on the opposite sides of the complexity spectrum. The strict organization holds minimum complexity, thus striving for a higher value for lowest project duration. Opposite can be said for the chaotic organization. The decentralized organization shows the optimal result for the quartile with the second lowest complexity. The decentralized organization has a medial complexity. This leads to the optimal project duration of the medial complexity quartile. The results show that a balanced organizational complexity delivers best results. Hence, reducing complexity is not necessarily beneficial.

In Figure 4, single factors are analyzed. The possibility of disobedience shows a project duration minimum around 20-25 %. The probability of decentralized decision making leads to a decrease of project duration up to a probability of approximately 30 %. After that threshold, the effect of a higher probability is receding. These two examples of single factor optimization show the potential of improving the setup of product development projects.

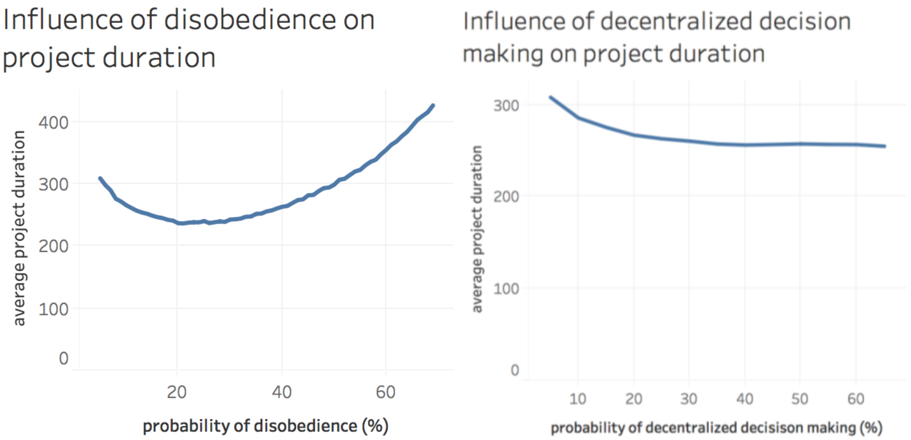


Figure 4. Influence of the probability of disobedience (left) and decentralized decision making (right)

These results deliver an answer on the second hypothesis. It can be shown that specific changes to a project can improve its outcome even though complexity rises. Thus, the least possible amount of complexity is not necessarily the optimal value. The results of Figure 4 show the potential for optimizing single factors of a project. Both analyzed factors increase the project complexity significantly, and up to a certain point, also improve project performance.

6 Conclusion

Organizational complexity influences the value of a product development project. Contrary to Birkinshaw and Heywood (2010); Carlucci et al. (2010); Danilovic and Browning (2007); Götzfried (2013); Lissack and Gunz (2005); Oehmen et al. (2015); Qureshi and Kang (2015), the increase of organizational complexity does not necessarily lead to a decrease of project time and cost. In a series of experiments, this paper suggests that there is an optimal value of organizational complexity for a project. Therefore, the increase of the right complexity drivers to reach that time and cost based project value leads to an improved project progression. Yet, it is also shown that the increase past this optimal point is harmful to the project. The simulation runs with different project scenarios and organizational behaviors all lead to different optima. Thus, not a general optimum for complexity can be stated but it is specific to each project. Both working hypothesis are therefore proven to be correct.

It is to be mentioned that this is not an exhaustive study. The goal of this paper is to present a different perspective on complexity. The presented findings demonstrate a unique approach to analyzing complexity from a different view point. Additionally, the simulative character of this paper does not enable a complete consideration of complexity. For example, the complexity through the human factor is not included. Results of this simulation can only be transferred carefully to real-life situations. Testing the simulated results in real organizations would be a next step in investigating the real-life project value of complexity.

References

Atkinson, R. 1999. Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International Journal of Project Management, 17*(6), 337-342.

Birkinshaw, J., & Heywood, S. 2010. Putting organizational complexity in its place. *McKinsey quarterly, 3*(6), 228.

Bonabeau, E. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences, 99*(suppl 3), 7280-7287.

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.

Carlucci, D., Lerro, A., & Skaržauskienė, A. 2010. Managing complexity: systems thinking as a catalyst of the organization performance. *Measuring business excellence*, *14*(4), 49-64.

Clark, K. 1989. Project scope and project performance: the effect of parts strategy and supplier involvement on product development. *Management Science, 35*(10), 1247-1263.

Danilovic, M., & Browning, T. R. 2007. Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Project Management, 25*(3), 300-314.

Dooley, K. 2002. Organizational complexity. *International encyclopedia of business and management, 6*, 5013-5022.

Efatmaneshnik, M., & Ryan, M. J. 2015. *Fundamentals of System Complexity Measures for Systems Design.* Paper presented at the INCOSE International Symposium.

Eppinger, S. D. 2002. Patterns of product development interactions.

Eppinger, S. D. 2009. *Integrating the Product, Process, and Organization Views of Complex System Development.* Paper presented at the 11th International DSM Conference. Munich: Hanser.

Eppinger, S. D., & Browning, T. R. 2012. *Design structure matrix methods and applications*.

Eppinger, S. D., Whitney, D. E., Smith, R. P., & Gebala, D. A. 1994. A model-based method for organizing tasks in product development. *Research in Engineering Design, 6*(1), 1-13.

Epstein, J. M. 1999. Agent-based computational models and generative social science. *Complexity, 4*(5), 41-60.

Epstein, J. M., & Axtell, R. L. 1996. Growing artificial societies: Social science from the bottom up (complex adaptive systems).

Gilbert, N., & Troitzsch, K. 2005. *Simulation for the social scientist*: McGraw-Hill Education (UK).
Götzfried, M. 2013. *Managing Complexity Induced by Product Variety in Manufacturing Companies: Complexity Evaluation and Integration in Decision-Making.*
Kerzner, H. 2013. *Project management: a systems approach to planning, scheduling, and controlling*: John Wiley & Sons.
Lindemann, U. 2006. *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden*: Springer.
Lissack, M. R., & Gunz, H. P. 2005. *Managing complexity in organizations: A view in many directions*: IAP.
Lissack, M. R., & Letiche, H. 2002. Complexity, emergence, resilience, and coherence: Gaining perspective on organizations and their study. *Emergence, A Journal of Complexity Issues in Organizations and Management, 4*(3), 72-94.
Luna, R., & Eppinger, S. D. 2015. *Structuring a Product Development Organization Based on the Product Architecture and Communication.* Paper presented at the DSM 2015: Modeling and managing complex systems-Proceedings of the 17th International DSM Conference Fort Worth (Texas, USA), 4-6 November 2015.
Macal, C. M., & North, M. J. 2006. *Tutorial on agent-based modeling and simulation PART 2.* Paper presented at the Simulation Conference, 2006 Proceedings of the Winter.
Macal, C. M., & North, M. J. 2010. Tutorial on agent-based modelling and simulation. *Journal of simulation, 4*(3), 151-162.
Masad, D., & Kazil, J. 2015. *Mesa: An Agent-Based Modeling Framework.* Paper presented at the 14th PYTHON in Science Conference.
Morelli, M. D., Eppinger, S. D., & Gulati, R. K. 1995. Predicting technical communication in product development organizations. *IEEE transactions on engineering management, 42*(3), 215-222.
Novak, S., & Eppinger, S. D. 2001. Sourcing by design: Product complexity and the supply chain. *Management Science, 47*(1), 189-204.
Oehmen, J., Thuesen, C., Ruiz, P. P., & Geraldi, J. (2015). *Complexity Management for Projects, Programmes, and Portfolios: An Engineering Systems Perspective*. Retrieved from
Patzak, G. 1982. *Systemtechnik-Planung komplexer innovativer Systeme: Grundlagen, Methoden, Techniken*. Berlin: Springer.
Ponn, J., & Lindemann, U. 2011. *Konzeptentwicklung und Gestaltung technischer Produkte: systematisch von Anforderungen zu Konzepten und Gestaltlösungen*: Springer-Verlag.
Qureshi, S. M., & Kang, C. 2015. Analysing the organizational factors of project complexity using structural equation modelling. *International Journal of Project Management, 33*(1), 165-176.
Rebentisch, E., Schuh, G., Sinha, K., Rudolf, S., Riesener, M., Mattern, C., & Stracke, F. 2016. *Measurement of Organizational Complexity in Product Development Projects.* Paper presented at the Management of Engineering and Technology (PICMET), 2016 Portland International Conference on.
Rouse, W. B., & Boff, K. R. 2005. *Organizational simulation* (Vol. 44): John Wiley & Sons.

Schwandt, A. 2009. Measuring organizational complexity and its impact on organizational performance–A comprehensive conceptual model and empirical study.

Sinha, K. 2014. *Structural complexity and its implications for design of cyber-physical systems.* Massachusetts Institute of Technology,

Sinha, K., & de Weck, O. L. 2013. *Structural complexity quantification for engineered complex systems and implications on system architecture and design.* Paper presented at the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.

Sosa, M. E., Eppinger, S. D., & Rowles, C. M. 2004. The misalignment of product architecture and organizational structure in complex product development. *Management Science*, *50*(12), 1674-1689.

Steger, U., Amann, W., & Maznevski, M. L. 2007. *Managing complexity in global organizations* (Vol. 5): John Wiley & Sons.

Sturtevant, D. J. 2013. *System design and the cost of architectural complexity.* Massachusetts Institute of Technology,

Turner, J. R. 2008. *Handbook of project-based management*: McGraw-Hill Professional Publishing.

Vidal, L.-A., Marle, F., & Bocquet, J.-C. 2011. Measuring project complexity using the Analytic Hierarchy Process. *International Journal of Project Management, 29*(6), 718-727.

Weber, C. 2005. *WHAT IS “COMPLEXITY”?* Paper presented at the DS 35: Proceedings ICED 05, the 15th International Conference on Engineering Design, Melbourne, Australia, 15.-18.08. 2005.

Contact: Nikolaus von Bary, Technical University Munich, Laboratory for Product Development and Lightweight Design, Boltzmannstr. 15, 85748 Garching, Germany, nikolaus.bary@tum.de

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Experimenting with the NK and DSM Models

Reem Ali Ahmad, Ali Yassine
Department of Industrial Engineering & Management,
American University of Beirut, Beirut, Lebanon

Abstract: The theory of complex systems, which has been applied successfully in evolutionary biology, is gaining popularity for the modeling and analysis of complex product development (PD) systems. Modeling complex PD systems is essential to understand how system elements and their dependencies impact system properties in several aspects such as performance, convergence, and evolution. In this paper we use the NK and NKC models to simulate and analyze complex PD systems, which are represented by the design structure matrix (DSM). The main objective is to assess whether these models can be useful in analyzing DSMs; particularly, assessing the effect of architecture on performance and evolution.

Keywords: NK Model, Design Structure Matrix (DSM), Product Development, Complex Systems, Performance Evaluation

1 Introduction

The theory of complex systems, which has been applied successfully in evolutionary biology (to study the dynamics and evolution of biological systems), is gaining popularity in product development (PD) to model and analyze man-made systems (e.g., Frenken and Mendritzki, 2012; Oyama et al., 2015). In fact, the biological domain is considered an analogy to complex PD systems where the genes in biological organisms correspond to the components in complex PD systems and genes in biological organisms depend on each other in a similar way to the components in man-made systems. Complexity of biological organisms is reflected by the dependencies between the genes; that is, when one gene is mutated, it may not just affect its own functionality but also affects the functionality of all other interdependent genes (Frenken, 2006). The main difference between the two systems is that man-made systems are designed by designers who are responsible for making the design decisions whereas biological systems depend on natural selection (Beesemyer et al., 2011).

This analogy between biological organisms and man- made complex systems is valid in terms of product evolution as well. Products evolve throughout the generations due to the continuous changes in the interdependent components' design, which increases the systems' performance. It has been argued that the way these interdependencies are distributed between the system's components (i.e. product architecture) affects the product's performance and evolvability (Rivkin and Siggelkow, 2007; Luo, 2015). In this context, modeling complex PD systems is essential to understand how the system elements and their dependencies impact system properties in several aspects such as performance, cost, improvement, convergence, and evolution.

According to the NK model, a product system can be defined as a complex system consisting of a set of N components (or modules), each of which is intended to deliver a specific functionality (Kauffman, 1993). Hence, each component delivers a specific function and, in turn, contributes some value to the overall product system. This value is referred to by the performance or the fitness value of the component. This component's performance depends on its own (design) decision and the decisions of one or more other components (depending on the system architecture). The decisions made at the component level are binary. That is, each component is available in two variants, which represent two alternative designs. A complete product contains exactly one variant of each component. A vector of length N whose i^{th} element represents a variant of the i^{th} component is called a design configuration. Standard practice in the NK literature denotes the variants by 0 and 1, which allows a configuration to be represented by a binary string (e.g., 0010 for a vector of length $N = 4$). The N-dimensional possibility space is called the design space and a specific component configuration defines a product design. Moreover, a complete product has a corresponding product (system) fitness that depends on the fitness values of its components. The actual resemblance of this product fitness is a measure of the performance of the system as a whole. For example, if the system is a team of employees, then the fitness of the system resembles the problem-solving effectiveness of this team (Solow et al., 2000).

In this paper we introduce an NK-based simulation model to analyze the design structure matrix (DSM) to assess the effect of the product architecture on product performance. In the next section, we introduce the basics of the NK model, and then we test its behavior based on varying N and K values. In Section 3, we introduce the notion of NK model using sub-blocks. In Section 4, we introduce the NKC model and run tests to compare its performance to the standard NK model. We test the various NK models on a set of different systems architectures in Section 5. We conclude the paper in Section 6.

2 NK Model Fundamentals

In the NK model we consider a system of N components, where each component depends on K other components (Kauffman, 1993). The NK Model is a mathematical representation of these dependencies, i.e. it assigns to each component a mathematical measure that represents the component's fitness value, taking into account the dependencies between components, as will be explained later in this section. To apply the NK model to the N size system, a N size Design Structure Matrix (DSM) is used to model and represent the system and its components' dependencies, as illustrated in Figure 1.

Suppose we have 3 components in a system, where the performance of each component depends on its own (design) decision and on the decisions of other components. In this case, N=3 and K=1.

	1	2	3
1	■	X	
2		■	X
3	X		■

Figure 1: DSM Representation of a Complex System

Figure 1 represents the scenario where each off-diagonal mark "X" represents a dependency between two components (Yassine and Braha, 2003). For example, the DSM

(assuming that row i depends on column j) shows that the performance of component 1 depends on its own (design) decision and the decisions of component 2. Similarly, component 2 depends on component 3, and component 3 depends on component 1.
The NK model starts by randomly assigning to each of the *N* components discrete random states (either 0 or 1) and corresponding random fitness values sampled from a uniform distribution ranging between 0 and 1. The fitness of the system, call it F_1, is the average of the fitness values of the N components and can be calculated according to the formula in Equation (1).

$$F_1=\sum fi/N \quad (1)$$

Where f_i is the fitness value of component i. In our case, shown in Figure 1, *i* ranges between 1 and 3 ($1\leq i\leq 3$) since there are 3 components in the system.

Then, one of the *N* components is randomly chosen to change its state and its corresponding fitness value. Furthermore, we change the fitness values of all the components that depend on this chosen component. For example, if we choose to change the state of component *i* ($1\leq i\leq N$), then if its state is 0 it becomes 1 and vice-versa. Then, we change the fitness value of component *i* as well as the fitness values of all the components *j* ($1\leq j\leq N$) that depend on component *i*.

Finally, the average fitness is recalculated, to obtain a new average fitness, call it F_2. If F_2 is greater than F_1, then we repeat the above process starting with the new obtained string of states and their corresponding fitness values. If F_2 is less than F_1, then we repeat the above process after choosing a component other than one previously chosen. This simulation process continues until a maximum average fitness is reached. Note that if a string of states is revisited, then their corresponding fitness values should be retained.

2.1 NK Model Simulation

For the DSM in Figure 1, the NK model works as follows. After randomly initializing the states and the fitness values of these components, we obtain initial states 110 and their corresponding fitness values 0.85, 0.57 and 0.63, resulting in an initial average fitness F_1=0.68 (Refer to the 7th row in Table 1). Then, the third component is randomly chosen so its state changes from 0 to 1 and its corresponding fitness value as well as that of component 2 change to 0.02 and 0.55 respectively, resulting in the 8th row in Table 1.

Table 1: Enumeration of the fitness values of the DSM in Figure 1

	States	**f_1**	**f_2**	**f_3**	**F**
1	**000**	0.31	0.72	0.37	0.47
2	**001**	0.31	0.42	0.51	0.41
3	**010**	0.38	0.57	0.37	0.44
4	**011**	0.38	0.55	0.51	0.48
5	**100**	0.15	0.72	0.63	0.5
6	**101**	0.15	0.42	0.02	0.2
7	**110**	0.85	0.57	0.63	0.68
8	**111**	0.85	0.55	0.02	0.47

This iteration results in the new average fitness $F_2=0.47<0.68=F_1$. For this, we return to the initial '110' string states and randomly choose a new component, i.e. any component other than the 3rd component. This process is repeated until a maximum average fitness F_{max} is reached. Table 1 enumerates the total 8 cases of this DSM. It is worth noting that the fitness values are almost always between 0.5 and 0.7 since we are sampling from a Uniform distribution between 0 and 1.

2.2 Effect of Varying K on the fitness values in the NK Model

To study the effect of K on the evolution of the fitness values, we ran the NK model on three DSMs of size 5, but with different number of dependencies K: a) K=0, b) 0<K<N-1 and c) K=N-1. The variation of the fitness values in these 3 cases is shown in Figure 2.

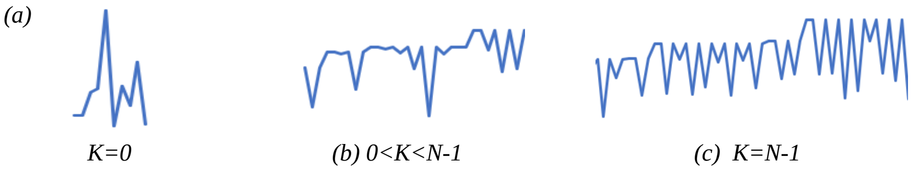


Figure 2: Evolution of the Fitness for Various Values of K (N=5)

Figure 2a represents the case where K=0, i.e. the system has no interactions among its components. In this case, there will only be one state (either 0 or 1) for each element that is responsible for making the highest fitness contribution to the system. This maximum fitness, i.e. the only global optimum, is represented by the highest single peak in Figure 2a. All other sub optimal fitness values will eventually reach the global optimum after having passed through all their neighboring states, which obviously have lower fitness than the global optimum.

We notice from Figure 2b that as the number of dependencies increases to take any value between 0 and N-1 (K=2 in our case), the number of fluctuations increases, and the graph becomes multi-peaked. In this case, each element depends on multiple other elements in the system, causing the number of the local optima to increase significantly and thus making it harder for each element to reach an optimum.

In the third case, as K reaches it maximum value, i.e. K=N-1 (K=4 in our case), the DSM become a completely rugged landscape where each element depends on all other elements in the system. This property causes the search process for the maximum fitness to be very difficult, as represented by the huge increase in the peaks of the graph, in Figure 2c.

2.3 Effect of N and K on the NK Model

To study the effect of the number of elements N and number of dependencies K on the system's behavior, the NK model is applied on several DSMs having different N and K. The corresponding changes in the fitness values and number of iterations are observed and shown in Figure 3.

Observation 1: As shown in Figure 3, for a fixed *N* (the number of components) and as *K* (the number of dependencies) increases, both the fitness and the number of iterations are not significantly affected. However, both the fitness and the number of iterations increase with *N* for a fixed *K*.

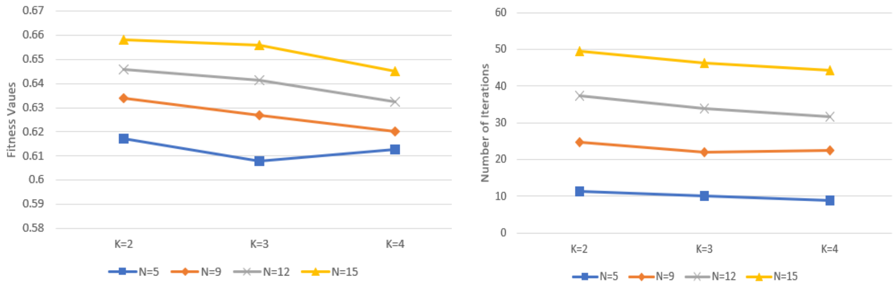
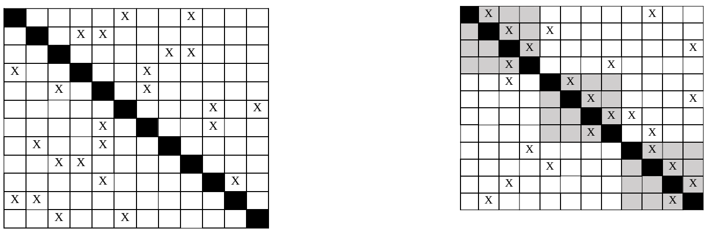


Figure 3: Effect of N and K on the DSM'S behavior

3 NK Model using Sub-blocks

The NK model is also applied in this section; however, the DSM is divided into sub-blocks prior to simulation. In this case, K is divided into two components; K_i and K_o, where K_i+K_o=K, K_i is defined as the number of dependencies within the same sub-block, and K_o as the number of dependencies outside the sub-block. For example, consider Figure 4b, where a DSM of size 12 and K=2 (K_i =1 and K_o=1), is divided into three sub-blocks of four components each.

Both Figures 4a and 4b have the same number of components N and dependencies K; however, the main difference is the way these dependencies are distributed. In Figure 4a, interactions between components are randomly distributed, however, in Figure 4b, they are classified according to the number of dependencies within and outside each sub-block, as described above. For example, the first DSM row has 2 marks (i.e. K=2). One of these marks is within the first block in the grey part of the row (since K_i =1) and the other mark is within the white part of the first row (since K_o=1). The rest of the marks are similarly allocated for each row in the DSM.

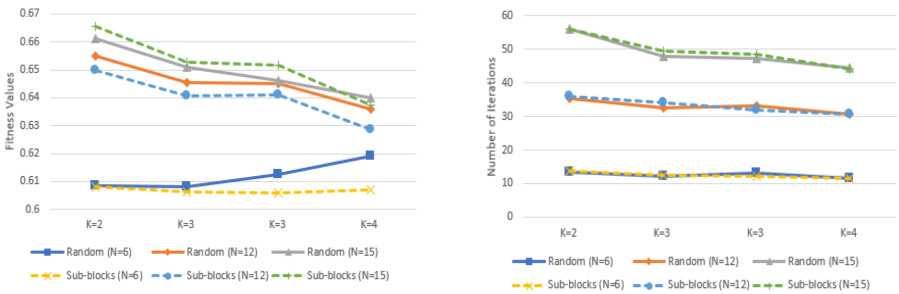


(a) 12 sized Random DSM (b) 12 sized DSM with sub-blocks

Figure 4: Sample of 12 sized DSMs having different dependencies' distribution

3.1 Effect of N and K on Random NK model and NK model with Sub-blocks

To study the behavior of the DSM with sub-blocks and test how it differs from the random DSM, the NK model is tested for 200 runs on both random and sub-blocks DSMs. This test is applied on DSMs with different number of components (N=6, 12 and 15) and dependencies (K=2, 3 and 4) to compare the maximum average fitness values and the average number of iterations executed by different cases. Also, note that two cases have been considered for K=3; either K_{in} =1 and K_{out} =2 or K_{in}=2 and K_{out}=1.



(a) Variation of the fitness Values (b) Variation of the number of iterations

Figure 5: Variation of the fitness and number of iterations of the random and sub-blocks DSMs as a function of K

Observation 2: As shown in Figure 5, both random and sub-blocks DSMs behave similarly with an increase in K. We can conclude that the effect of distributing the dependencies between the components using sub-blocks is almost negligible on the system's fitness and number of iterations.

3.2 NK Models with Different Numbers of Sub-blocks

Each DSM, with a certain number of components N, can be divided into different number of sub-blocks. For example, the 12 sized DSM, shown in Figure 4b, is divided into three sub-blocks of 4 components each; however, it can be divided into 2 sub-blocks of 6 components each, or into 4 sub-blocks of 3 components each, etc. Accordingly, we tested the NK model on a 12 sized DSM, with K=4, divided into different number of sub-blocks to study the effect on the fitness values and the number of iterations. We observed that changing the number of sub-blocks did not significantly impact the fitness values nor the number of iterations.

4 NKC Model Fundamentals

In the NKC model, we consider a system of size *N*, but with *S* subsystems (Hordijk and Kauffman, 2005), where:

- N: number of total components that are distributed along S sub systems
- K: number of inter dependencies inside the sub system
- C: number of external dependencies, that is each component in each sub system depends on C other components from other sub systems
- Nj': number of components inside subsystem j. Note that the number of components within a one subsystem may differ from the number of components in another subsystem

We start by randomly assigning discrete random states (either 0 or 1) and random fitness values sampled from a Uniform distribution ranging between 0 and 1 to all components in all subsystems. Then, a random subsystem *j* ($1 \leq j \leq S$) is selected and a component *i* from subsystem *j* is randomly chosen to change its state and its corresponding fitness value. Next, we randomly sample for the fitness of all components in subsystem *j* that depend on component *i*. The average fitness of subsystem *j* is calculated as follows in Equation 2:

$$F_j = \sum fi / N_j' \quad (2)$$

where f_i represent the fitness value of component *i* in subsystem *j*.

If the new average fitness of subsystem *j* is greater than the previous average fitness, then we sample for the fitness values of components, in subsystems other than subsystem *j*, which depend on component *i*. While if the new average fitness of subsystem *j* is lower than the previous average fitness, we chose another component from subsystem *j*.

These steps are repeated until a maximum average fitness of subsystem j is reached. After applying the above scenario for all subsystems S, the maximum average fitness of the whole system is calculated as: $F = \sum F_j / S \quad (3)$

4.1 Examining the Difference between NK and NKC Models

To notice the difference between the NK and NKC models, both models were run in parallel for 1000 runs on the 12 sized DSM, represented in Figure 4b. The variables of this DSM, represented in both NK and NKC models, are shown in Table 2.

Table 2: Variables of the DSM in Figure 4b in NK and NKC Models

	NK Model	**NKC Model**
N	12	12
K	2	1
C	0	1
S	1	3
N'	-	4

The average fitness values and average number of iterations of the 1000 runs are recorded in Table 3. The simulated 1000 maximum fitness values and number of iterations are displayed in Figure 6.

Table 3: Average maximum fitness and number of iterations in along the 1000 runs

	NK Model	NKC Model
Average Maximum Fitness	0.6505	0.5874
Average Number of Iterations	32.99	24.468

Observation 3: As shown in Table 3 and Figure 6, the NKC model reach a lower maximum fitness, on average, than the NK model and at a lower average number of iterations as well. However, along the 1000 runs, there is not a clear relation between the maximum fitness of the NK and NKC models in each run. As for the number of iterations, the NKC model clearly takes less iterations than the NK model, almost throughout all the 1000 runs, as shown in

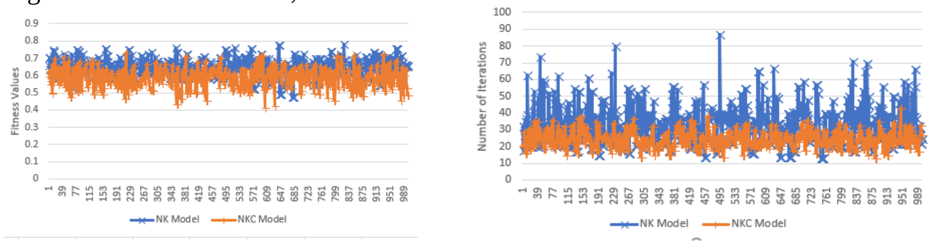
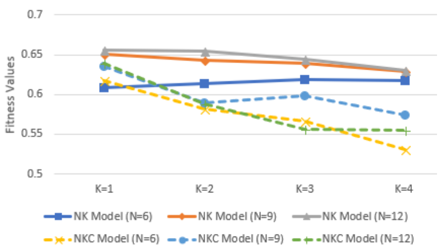


Figure 6 (right).

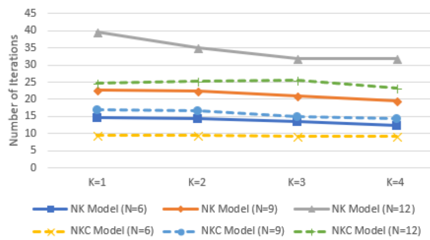
Figure 6: Fitness Values (Left) and Number of Iterations (Right) in NK & NKC Models

4.2 Effect of N and K on the Performance of NK and NKC models

To test the effect of changing N and K on the system's performance in each of the NK and NKC models, both models are applied on DSMs of different sizes (N=6, 9 and 12) and different dependencies (K=1, 2, 3 and 4). Note that changing the number of dependencies in the NKC model is done by either increasing the number of internal dependencies K or the number of external dependencies C.



(a)Variation of the average maximum fitness



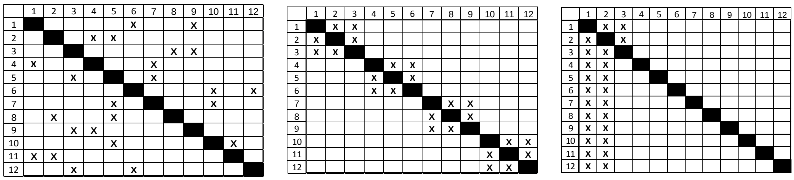
(b)Variation of the avg. no. of iterations

Figure 7: Variation of the average max. fitness as a function of K in the NK & NKC models

Observation 4: As shown in Figure 7, as the number of dependencies K increase, the (average) maximum fitness of both the NK and NKC models generally decrease. In the NK model, the decrease occurs slowly as K increases and the curve somehow remains flat, however the fitness in the NKC model decreases at a faster rate, and this is clear from the slopes that appear to be steeper in the NKC model.

5 Effect of N, K and Architecture on NK and NKC Models

In this section, we perform a comprehensive study in which we test both, the NK and NKC models, on different numbers of elements N (12 and 16), different number of dependencies K (1,2 and 3) and different architectures (Random, Block-Diagonal, and Centralized).

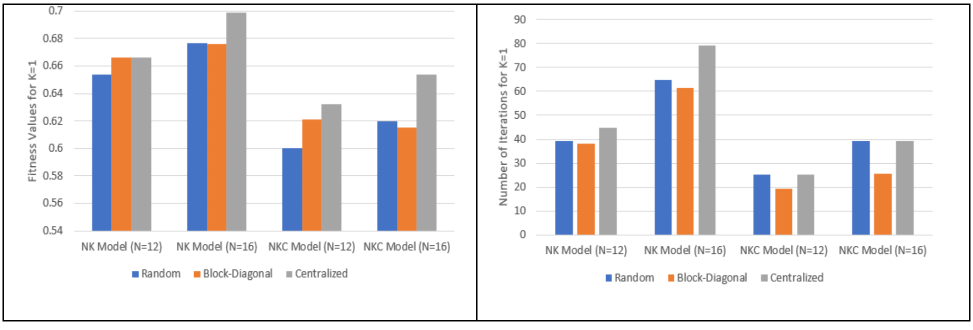


(a) Random *(b) Block Diagonal* *(c) Centralized*

Figure 8: Sample DSM architectures (N=12, K=2)

Sample DSMs of the different architectures is shown in Figure 8. The results of this test (fitness and number of iterations), for each of the three architectures, in the NK and NKC models are presented in Figure 9.

Observation 5: When comparing the fitness values of the DSMs of different architectures in Figure 9 (left-side panels) it is noticed that the Random DSM almost has the lowest fitness values for both values of N=12 and N=16 in the NK and NKC models. On the other hand, we can see that the Centralized DSM always has the highest fitness values. As for the Block-Diagonal DSM, its fitness values vary between those of the Random and Centralized DSMs.



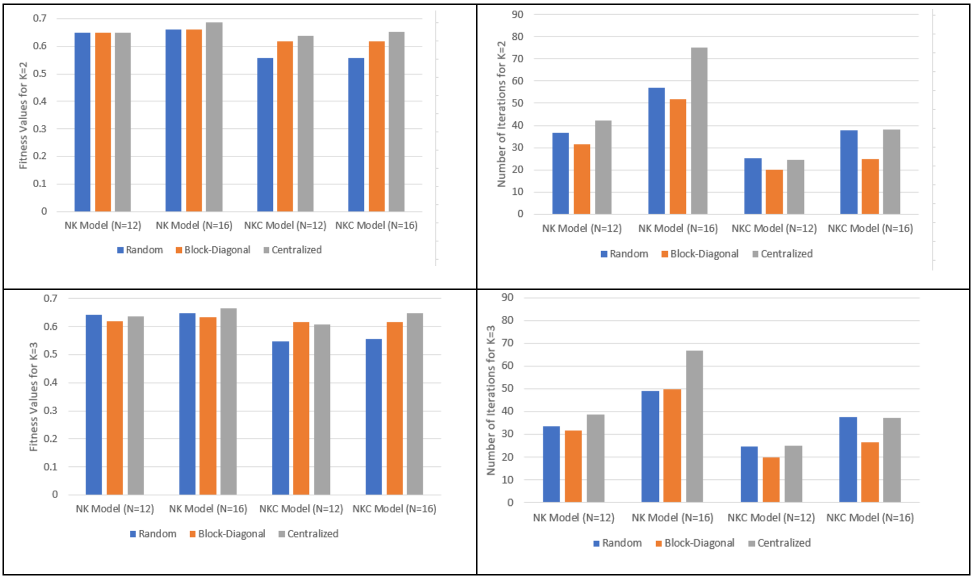


Figure 9: Variation of the fitness values (left-side panels) and the number of iterations (right-side panels) as a function of N in NK and NKC models for different DSM architectures

In Figures 9 (right-side panels), we can see that the Centralized DSM execute the highest number of iterations, whereas the Block-Diagonal takes the lowest number of iterations for both values of N=12 and N=16 in the NK and NKC models. It is noticed that the difference in the number of iterations executed between the three architectures increase in each of the NK and NKC models as N increases, i.e. the difference in the number of iterations between the three architectures is greater when N=16 than when N=12. Also, when comparing the variation of the number of iterations for the different values of K (Figures 9 (b), (d), (f)), it is noticed that the behavior and pattern of variation is the same.

6 Summary and Conclusion

In this paper, we experimented with the NK and NKC models to investigate their utility in the analysis of PD systems represented by DSM models. We tested various parameters (in the NK model) that may impact the system's performance evolution, mainly the number of components in the system, the number of dependencies between these components, and the system's architecture.

We found that as K increases, the fitness is mostly unaffected; however, the process of searching for the maximum fitness becomes harder due to having multiple local optima (when 0<K<N-1) rather than one global optimum (when K=0). Also, as N increases, we noticed that the number of iterations increase, despite the number of dependencies K. As for the fitness, it increases with N, provided that we are comparing for the same value of K. Finally, we concluded that if the components randomly interact with each other, the system's fitness and number of iterations will be smaller than the case when the elements

depend on each other in a structured way (such as Block-Diagonal and Centralized DSMs).

The standard NK and NKC models can be useful for the analysis of PD systems only if some adjustments are made, which relate to the difference between biological systems and man-made (engineered) systems. First, performance in engineered systems is not random and should be proportional to allocated effort. Second, choice of the component to work on is also not random but a deliberate choice is made by the development team. Third, specifying the type of dependencies between the components; that is, when the performance of one component increases, the performance of its dependent components may increase or decrease depending on the nature of this dependency. Also, time and cost implications of the evolution process is not taken is not account. Finally, PD projects have a budget allocated to them and scheduled deadlines to meet. Hence, cannot evolve freely until maximum fitness is reached.

References

Beesemyer, J. Clark, et al., 2011. Developing methods to design for evolvability: research approach and preliminary design principles, 9th Conference on Systems Engineering Research.

Frenken, Koen, and Stefan Mendritzki., 2012. Optimal modularity: a demonstration of the evolutionary advantage of modular architectures. Journal of Evolutionary Economics 22, no. 5: 935-956.

Frenken, Koen., 2006. A fitness landscape approach to technological complexity, modularity, and vertical disintegration. Structural Change and Economic Dynamics17, no. 3: 288-305.

Hordijk, Wim, and Stuart A. Kauffman., 2005. Correlation analysis of coupled fitness landscapes. Complexity 10, no. 6: 41-49.

Kauffman, S.A., 1993. The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press.

Luo, Jianxi., 2015. A simulation-based method to evaluate the impact of product architecture on product evolvability. Research in Engineering Design 26, no. 4: 355-371.

Oyama, Kyle, Gerard Learmonth, and Raul Chao., 2015. Applying complexity science to new product development: modeling considerations, extensions, and implications. Journal of Engineering and Technology Management 35: 1-24.

Rivkin, Jan W., and Nicolaj Siggelkow., 2007. Patterned interactions in complex systems: Implications for exploration. Management Science 53, no. 7: 1068-1085.

Solow, D., Burnetas, A., Tsai, M. C., & Greenspan, N. S., 2000. On the expected performance of systems with complex interactions among components. *Complex Systems*, *12*(4), 423-456.

Yassine, Ali, and Dan Braha., 2003. Complex concurrent engineering and the design structure matrix method. Concurrent Engineering 11, no. 3: 165-176.

Contact: **Professor Ali A. Yassine,** American University of Beirut, Department of Industrial Engineering & Management, Hamra-Bliss Street, PO Box 11-0236, Beirut, Lebanon, 00-961-1-350-000 Extn. 3439, ali.yassine@aub.edu.lb

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Clustering Organization Structure in Product Development Projects Using Similarity

Na Yang[1], Qing Yang[1], Tao Yao[2]

[1]School of Economics and Management, University of Science & Technology Beijing, China
[2]The Harold & Inge Marcus Department of Industrial & Manufacturing Engineering, Pennsylvania State University, USA

Abstract: A product development (PD) project is a complex social network, in which teams have their own attributes and are related by information flow. Similar team attributes and the complex patterns of technical dependency among teams both affect organization modularity. This paper provides an innovative spectral clustering approach that merge team attributes and relationship of teams. To measure the similarity of PD teams, we analyze the similar attributes of team and build structural models to capture the technical communication dependency among teams via the product-organization multi-domain matrix (MDM). We use two metrics to evaluate the clustering solutions and confirm that the proposed approach provides effective reduction of PD coordination complexity.

Keywords: product development, organization design, design structure matrix (DSM), similarity, spectral clustering

1 Introduction

A key managerial issue in product development (PD) is how to establish an effective organization architecture, because the complexity of interactions among which may reduce efficiency and introduce additional risks (Yang et al., 2014). A common but challenging objective in organization architecting concerns modularity—i.e., parsing the set of organizational elements (e.g., teams or individuals) into subsets, groups, or modules, such that the elements' relationships within each group are much stronger than those across groups (Tripathy and Eppinger, 2013). Many prior studies have applied some kind of clustering algorithm to optimize a model of the organization architecture, such as an organization design structure matrix (org DSM) (Tripathy and Eppinger, 2013; Yang et al., 2014).

Classical clustering algorithms are popular. For example, k-means (Ahmad & Hashmi, 2016) are based on the node attributes, while Fast-Newman algorithm (Newman, 2004) focuses on relationship. Most of clustering algorithms separate the attribute and relationship of nodes while clustering a complex graph. In fact, they both affect the results of modularity. For example, similar interest and friendship make two users close to each other in social network. Therefore, we aimed to formulate the DSM clustering problem combined the attributes and relationship of teams. Team attributes are based on social similarity with respect to significant background characteristics, such as race, sex and level of education et al. Teams who share important social characteristics are presumed to have common experiences, leading to shared knowledge.

Spectral clustering algorithm based on graph theory provides a stronger and more stable approach for finding the global optimum (Schaeffer, 2007; Sarkar et al., 2014), especially for non-convex datasets, and are well suited for application to real problems (Sarkar et al., 2014). The spectral clustering algorithm maximizes intra-cluster similarity and minimizes inter-cluster similarity. The similarity matrix is thus a critical input to a spectral clustering algorithm (Schaeffer, 2007). Many researchers have developed methods to measure similarity (Schaeffer, 2007).

Amount of research highlights the importance of similarity between teams or members for team process, such as team functioning and knowledge exchange. Larzarsfeld and Merton(1954) believes that interactions are more likely to occur between members or teams that are similar to each other. The similarity of knowledge bases inherent results in the recipient and partner team being more inclined to interact with one another and being able to understand the linkages between one another's knowledge stocks, which provides more favorable conditions for knowledge sharing. Therefore, the more similar team attributes are, the more intensive communication and interaction will be.

In this paper, we present an improved optimization approach, based on spectral clustering, that accounts for the similarity of teams in the PD organization.

2 An Improved spectral clustering for measuring modularity

It is important to take both attribute of teams and relationships between them into consideration. Thus we define a similarity matrix which merges team attribute and relationship. First, we analyze the similar characteristics of team, such as product-related expertise, process-related expertise and so on, which enhance the formation of relationships and interactions among them. Then, we infer technical communication strength among teams which reflect each team's role toward the design of components. Finally, we establish the cluster model of the graph containing both attribute of teams and relationship between them.

2.1 The attributes of the organization team

There are a lot of similar characteristics when selecting a cooperative team. For example, social-category similarity, work-style similarity, similar work habits and ethics (Zellmer-Bruhn et al., 2008) and so on. This paper examines two types of similarity attributes between teams− product-related expertise and process-related expertise.

2.1.1 Product-related expertise

Sosa (2011) defined product-related expertise that is associated with the specific functional and architectural attributes of the product under development. To collect data on areas of expertise, we ask them to indicate "the areas in which they considered themselves experts" based on what component they complete. Teams could select from n areas of product-related expertise which provided a more granular description of each area of expertise, was assembled by a technical product manager. The score of team's product-related expertise that is between *0* and *1* is ascertained by the project manager, design engineers, and other subject matter experts, according to their knowledge and experience, which reflects team members had expertise relevant to area of product-related expertise. The product-related expertise differential between team *i* and *j* can be calculated with equation (1):

$$M_{ij}^{PE} = \sqrt{\sum_{k=1}^{n} (P_{ik} - P_{jk})^2} \quad (1)$$

where P captures the team's product-related expertise technologies. Then we devised the expertise differential M_{ij}^{PE} based on the Euclidean distance between *i* and *j*.

2.1.2 Process-related expertise

Sosa (2011) defined process-related expertise that is associated with the procedures and activities associated with product development generally. For example, "process and product management," "product conception," "system design" and so on. The score of team's process-related expertise that is between *0* and *1* is the same as product-related expertise. The process-related expertise differential between team *i* and *j* can be calculated with equation (2):

$$M_{ij}^{TE} = \sqrt{\sum_{k=1}^{n} (T_{ik} - T_{jk})^2} \quad (2)$$

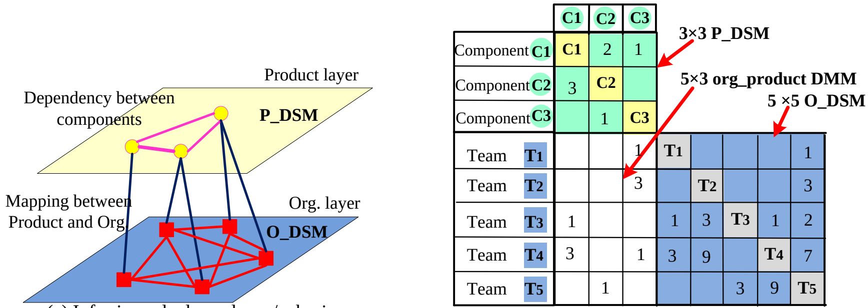
where T captures the team's product-related expertise categories. Then we devised the expertise differential M_{ij}^{TE} based on the Euclidean distance between *i* and *j*.

So, the total differences between team *i* and *j* can be calculated with Eq. (3), where ω_1, ω_2 are weight coefficients, $\omega_1 + \omega_2 = 1$. In this paper, we discuss only the case when $w_1 = w_2 = 0.5$.

$$M_{ij}^{TD} = \omega_1 M_{ij}^{PE} + \omega_2 M_{ij}^{TE} \quad (3)$$

2.2 Modeling the relationship between organization team via product-organization MDM

We adopt an approach, recently proposed by (Yang et al., 2014), to derive the technical dependency between teams in org DSM from an MDM model inclusive of a product DSM and an organization-product DMM, as shown in Fig. 1(b). In the upper-left of the MDM, product DSM *P_DSM* models the technical communication among teams at the component level, which reflects the roles of teams in the design process of components containing some functions and allows teams to maintain control over all the functions that perform related tasks. And in the lower-left of the MDM, $DMM_{OP}(i, I)$ models the degree of involvement (e.g., the consumed time) of team *i* in the design of component *I*. For example, the *P_DSM* (3, 2) is nonzero in the product DSM, which means the design of component C2 will directly impact C3. Further, from the column of DMM_{OP}, we find that teams T5 and T4 responsible for developing product components C2 and C3 respectively. Then, we can infer a dependency of T4 on T5 which reflects the direct role relationship between these teams in the designing process of components C2 and C3.



(a) Inferring role dependency/cohesion between teams from the product layer

(b) Org_product MDM

Fig.1. Modeling technical dependency among teams via MDM

So, we derive the technical dependency between teams in org DSM from the product DSM via the DMM (which are all part of the MDM in Fig. 1(b)). The org DSM, *O_DSM*(*i*,*j*), to the right of Fig. 1(b), reflects the integrated effects of the dependency relationships among the product components and the teams' degrees of involvement in the components' design. Hence, using *P_DSM* and DMM_{OP}, the technical communication strength between teams *i* and *j* is modeled as:

$$O\_DSM(i,j)=\sum_{I=1}^{p}(DMM_{OP}(i,I)\times\sum_{J=1,J\neq I}^{p}(DMM_{OP}(j,J)\times(P\_DSM(I,J)+P\_DSM(J,I))) \qquad (4)$$

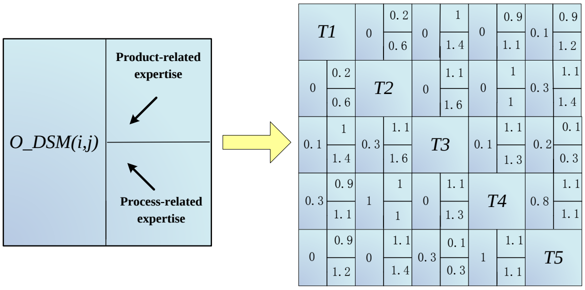
In this paper, the value of *P_DSM* and DMM_{OP} are evaluated by analyzing the functional dependency relationships among components and the team's involvement degree in the component's design, respectively, as ascertained by the project manager, design engineers, and other subject matter experts, according to their knowledge and experience. *P_DSM*(*I*, *J*) and $DMM_{OP}(i, I)$ model the relationship at four levels: 0 = none, 1 = weak/low, 2 = medium, and 3 = strong/high. We normalize *O_DSM* by dividing all cells by the maximum cell value, thereby bounding all values in *O_DSM* (*i*,*j*) in [0, 1].

2.3 Building the Similarity Matrix of PD Teams

The differences of attributes between team *i* and *j* is defined as M_{ij}^{TD}.All the relationship can be denoted by$O_DSM(i,j)$.In order to merge the attribute and relationship of teams, we define the similarity matrix containing both information (attribute and relationship) of the entire graph. Thus, for each pair of team *i* and *j*, $S_{ij}=sim(i,j)$, in which S represents the ultimate similarity matrix. In this experiment, on the base of data density (Yi Xu et al., 2018), the functions are defined as follows:

$$sim(i,j) = \alpha \times e^{-\frac{O\_DSM(i,j)+O\_DSM(j,i)}{\sum_{u\in\Gamma(i)}(O\_DSM(i,u)+O\_DSM(u,i))+\sum_{u\in\Gamma(j)}(O\_DSM(j,u)+O\_DSM(u,j))}} + (1-\alpha)\times e^{-\frac{1}{M_{ij}+1}} \quad (5)$$

where $\Gamma(i)$ means the set of adjacent teams of *i*, and α means the similarity coefficient which is usually set as 0.4(Yi Xu et al., 2018). Data Density methods discover dense regions in space, where objects are adjacent to each other and separate them from sparse regions.



	T1	*T2*	*T3*	*T4*	*T5*
T1	1	1.63	1.32	1.36	1.36
T2	1.63	1	1.27	1.31	1.31
T3	1.32	1.27	1	1.34	1.73
T4	1.36	1.31	1.34	1	1.26
T5	1.36	1.31	1.73	1.26	1

(c) *sim(i,j)*

Fig.2. An example of calculating the sim matrix

Fig. 2 provides an example of calculating the similarity between teams. Fig. 2(a) can be captured with Eqs. (1)-(4).

2.4 Spectral Clustering Approach

Spectral clustering techniques make use of the spectrum of the data's similarity matrix to perform dimensionality reduction before clustering the data in fewer dimensions. The similarity matrix is an input to spectral clustering and the optimal partition maximizes the similarity of elements in the cluster (or subgraph) while minimizing the similarity between elements in different clusters. Ng-Jordan-Weiss (NJW) algorithm (Ng et al., 2002), which utilizes the Laplacian matrix, a simple normalization of the similarity matrix to optimize the normalized cut criterion according to the eigenvectors associated with the largest eigenvalues. We apply the following NJW algorithm-based, normalized spectral clustering procedure (Ng et al., 2002) because of its more robust performance.
We use two metrics to evaluate the clustering solutions. First, we adapt the *numerical dependency density* (NDd) measure (Chen and Lin 2003), the ratio of the total interaction strength (TIS) of all (non-zero) elements outside the clusters to the total number of cells outside the clusters:

$$NDd = \frac{TIS}{cell\_out} \quad (6)$$

Second, we use the global Silhouette index of the clustering (Slobodan Petrovi´c, 2006), which measures the quality of clustering by calculating the distance between each cluster and the distance between each team in the cluster. The definition of Silhouette index is as follows:

$$S(k) = \frac{1}{k}\sum_{i=1}^{k}\left\{\frac{1}{m_i}\sum_{i=1}^{mi}\frac{b_j^i - a_j^i}{\max[a_j^i, b_j^i]}\right\} \qquad (7)$$

where

$$a_j^i = \frac{1}{m_i - 1}\sum_{k=1, k\neq j}^{mi} d(T_j^i, T_k^i),\ j = 1, ..., mi.$$

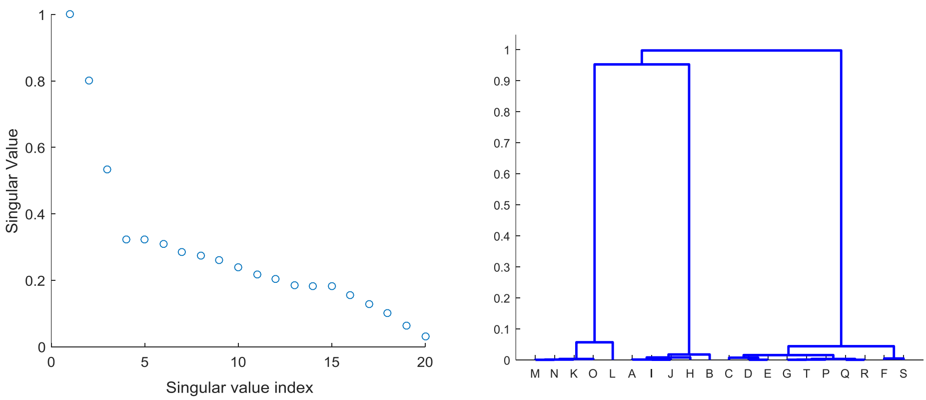
$$b_j^i = \min_{n=1,...,k; n\neq i}\left[\frac{1}{m_n}\sum_{k=1}^{m_n} d(T_j^i, T_k^n)\right], j = 1, ..., mi.$$

where $O = \{C_1, C_2, ..., C_k\}$ is its clustering into k clusters, $d(T_k, T_l)$ is the distance between T_k and T_l, $C_i = \{T_1^i, ..., T_{mi}^i\}$ is the i-th cluster, $i = 1, ..., k$ and $m_i = |C_i|$. The global silhouette take values between -1 and 1, the maximum value of which indicate the best clustering result.

3 Case Studies

We applied the proposed concepts and models to a PD project in an IT company involving 20 teams and 18 components. Based on the responses and other information provided, we built the product DSM, and the product-organization DMM. *P_DSM*(*I*, *J*) are measured by the added cost on component *I* when component *J* is designed or redesigned and *DMM*(*i*, *I*) are measured by the time required of team *i* in the design of component *I*.

First, using equation (4), we derived the technical communication/dependency strength among the teams. Next, we calculated the similarity matrix with equations (1)-(5) and applied the spectral clustering procedure in the Matlab® 15 software.



(a) (b)

Fig.3. Results of singular value and cluster tree using spectral clustering

Fig. 3(a) shows the sigular values for the similarity matrix, which is composed by the attributes and relationship of organizaion teams. 3 large singular values appear, which signals the appearance of 3 modules in the organization. Sarkar(2014) found that the number of outlying eigen or singular values, separated from the bulk of the spectrum, provides a good estimate of the actual number of modules in the system.
Fig. 3(b) shows the results of the modularity analysis: group 1 from teams 13 to 12(i.e., G1 [M, N, K, O, L]), group 2 from teams 1 to 2(i.e., G2 [A, I, J, H, B]), group 3 from teams 3 to 19(i.e., G3 [C, D, E, G, T, P, Q, R, F, S]).

	M	N	K	O	L	A	I	J	H	B	C	D	E	G	T	P	Q	R	F	S
M	M	.6	.6	.5	.1		.5													
N	.6	N	.5	.3	.6															
K	.5	.5	K	.3	.6					.3		.2								
O	.7	.3	.7	O	.5						.1									
L	.5	.4	.5	.4	L	.1		.2												
A					.1	A	.1	.5	.1	.7	.1	.1		.1					.1	
I	.5					.9	I	.2	.5	.1										
J					.4	.5	.2	J	.6											
H						.1	.5	.6	H	.1										
B			.3			.7	.1			B	.1	.9		.1						
C		.3				.1				.5	C	.3	.5	.6	.3	.3			.1	
D						.1					.3	D	.4	.9					.9	
E											.1	.4	E	.3					.2	
G						.1				.1	.1	.9	.3	G					1	
T											.5				T	.5	.8	.3		
P											.3					P				
Q											.5			.1	.8	.5	Q	.5		.8
R											.5				.3	.6	.5	R		.3
F						.1					.1	.9	.2	1					F	
S											.6				.8	.8	.3			S

Fig.4. Clustered O_DSM

Fig. 4 shows the resulting, clustered O_DSM that teams with high similarity (i.e., strong information exchange) are brought together in groups while connections between groups become weaker, thereby reducing the coordination challenges.
The Ndd of our proposed spectral clustering method is 0.022 and the Silhouette index of our method is 0.5323, which indicate the clustering result is well.

4 Conclusions

This paper provides a framework that enables managers to design a PD organization that can be coordinated more efficiently and effectively. The proposed approach of constructing the similarity avoids the use of Radial Basis Function, imports similar team attributes and the directed relationship into the similarity matrix.
The main limitations of this research are: how to quantify team attributes is very difficult; benchmarking our method against other clustering methods when it is very difficult to judge which one is the best (e.g., the applied situation may vary) and obtain (or reproduce) their programs.
Several aspects of the model presented in this paper merit further examination in future research. First, from the experiments, the attributes of teams can greatly affect the clustering results. There probably exist more factors we have not considered. Second, other data collection methods and dependency measurement methods theory that reduce the ambiguity of respondents' judgments.

Acknowledgments

This study was supported by the National Science Foundation of China (No. 71472013, No. 71528005 and No. 71872011)

Reference

Ahmad, A., & Hashmi, S. (2016). K-harmonic means type clustering algorithm for mixed datasets. *Applied Soft Computing, 48*, 39-49.

Chen S. J., L. Lin., 2003. Decomposition of interdependent activity group for concurrent engineering. Computers & Industrial Engineering, 44(3):435-459.

Lazarsfeld, P. F., & Merton, R. K., 1954. Friendship as a social process: A substantive and methodological analysis. Freedom and control in modern society, 18(1), 18-66.

Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys, 69*(6 Pt 2), 066133.

Ng, A. Y., M.I. Jordan, Y. Weiss., 2001. On spectral clustering: analysis and an algorithm. Proceedings of Advances in Neural Information Processing Systems14, pp. 849-856. MIT Press, Cambridge

Petrovic, S., 2006, October. A comparison between the silhouette index and the davies-bouldin index in labelling ids clusters. In Proceedings of the 11th Nordic Workshop of Secure IT Systems (pp. 53-64).

Sarkar, S., A. Dong, J. A. Henderson, P. A. Robisnon., 2014. Spectral characterization of hierarchical modularity in product architectures. Journal of Mechanical Design, 136(1):249-256.

Schaeffer, S. E., 2007. Survey: graph clustering. Computer Science Review, 1(1): 27-64.

Sosa, M. E. (2011). Where do creative interactions come from? The role of tie content and social networks. *Organization Science, 22*(1), 1-21.

Tripathy, A., S.D.Eppinger., 2013. Structuringbwork distribution for product development organizations. Production & Operations Management, 22(6): 1557-1575.

Xu, Y., Zhuang, Z., Li, W., & Zhou, X., 2017. Effective community division based on improved spectral clustering. Neurocomputing.

Yang, Q., T. Lu, T. Yao, B. Zhang., 2014. The impact of uncertainty and ambiguity related to iteration and overlapping on schedule of product development projects. International Journal of Project Management, 32(5):827–837.

Zellmer-Bruhn, M. E., Maloney, M. M., Bhappu, A. D., & Salvador, R. B., 2008. When and how do differences matter? An exploration of perceived similarity in teams. *Organizational Behavior and Human Decision Processes*, *107*(1), 41-59.

Contact: Qing Yang, University of Science & Technology Beijing, School of Economics and Management, No.30 Xueyuan Road, Haidian District, 100083 Beijing, China, 13911463353, yangqing@manage.ustb.edu.cn.

Part III: Product & System Architecture

DSM Modeling and Requirement Specification in Developing a Product Platform for Locks

T. Wilschut[1], L.F.P. Etman[1], J.E. Rooda[1], J.A. Vogel[2]

[1]Eindhoven University of Technology, 5600 MB, Eindhoven, The Netherlands
[2]Rijkswaterstaat, 3526 LA, Utrecht, The Netherlands

Abstract: This paper presents dependency structure matrix (DSM) modeling and requirement specification methods that support the development of a lock product platform. The study concerns methods for similarity, modularity, and commonality analysis of a navigation lock portfolio and the creation of a language for writing concise and unambiguous function- and design-specifications. In this paper, we present the methods that we have developed and show how we have used them for the development of a navigation lock product platform. The study bridges DSM modeling with requirements specification in an engineering systems design context.

Keywords: product platform, product family, similarity, modularity, commonality, specification language, requirement specification

1. Introduction

Navigation locks are vital assets in the Dutch infrastructure, which regulate the flow of water through the waterways and enable ships to cross differences in water levels between waterways. In the Netherlands, a considerable number of navigation locks were built during the first half of the previous century. In the coming decades, approximately fifty navigation locks have to be thoroughly renovated or replaced, since they have reached their end-of-life, no longer meet modern-day safety standards, or have insufficient capacity to keep up with growing waterborne transportation.

Historically, locks have been built using an Engineer-to-Order (EtO) production strategy. Each lock has been uniquely designed to meet location specific requirements and constraints. As a consequence, a great variety of lock designs currently exists in the Netherlands.

Lock asset managers have observed that due to the design variety, specialized knowledge, equipment, and spare parts are required to operate and to maintain the locks. The asset managers consider this to be inefficient and expensive. What is more, an EtO strategy requires excessive (human-) resources to renovate and to replace fifty locks within a few decades. Therefore, Rijkswaterstaat (RWS), the executive branch of the Dutch Ministry of Infrastructure and Water Management, founded the Multi-Water-Werk (MWW) project, which is dedicated to the modularization of locks, and the standardization of selected lock modules. By doing so, RWS aims to increase lock reliability and availability (RA), to decrease life-cycle-costs (LCC), and to decrease uncertainty in construction costs and time.

Design and realization of a series of locks using a modularized architecture and standardized solutions for selected modules, resembles a mixture of a Make-to-Order (MtO) and a Configure-to-Order (CtO) production strategy. An MtO strategy requires a basic product structure (design) to be present at the moment a customer order is received, i.e., in the case of RWS at the moment a lock is due for renovation or replacement. This basic product structure is subsequently modified to specific customer needs. A CtO strategy requires standard module and component designs to be present at the moment a customer order is received. A selection of standard modules and component designs is subsequently combined and configured to customer specific needs. A CtO approach allows for mass customization while still benefiting from economies of scale (Jiao et al., 1999).

A challenge in implementing MtO and CtO production strategies is to balance the product variety that is offered to the customer with the internal complexity of managing the design of many product variants (Jiao et al., 2007). To do so effectively, companies often resort to the creation of a product platform, which is defined by Meyer (1997) as: '*a set of subsystems and interfaces developed to form a common structure from which a stream of derivative products can be efficiently developed and produced.'*

The level of standardization of a product platform may differ. Alblas et al., (2012, 2014), for example, advocate the usage of function - technology platforms in traditional EtO industries. Such a platform contains a standard set of functions, working-principles, and technologies from which engineers can choose during the conceptual and embodiment design phases of a design project. It does not contain detailed designs of standard components.

This study contributes to the development of a lock product platform composed of fully-, semi-, and non-standardized component modules and the interfaces between them. The platform distinguishes between basic modules and optional modules. Basic modules are groups of components that are always present in all locks. Optional modules are groups of components that are only occasionally present in a lock. The level of standardization of each module may range from a functional level to a full detailed design level. RWS can use this platform for the efficient development of (semi)-standardized locks that meet location-specific requirements and constraints while reducing the design variety in their lock portfolio.

2. Research objectives

The objectives of this research are two-fold. Firstly, methods are sought to study the similarity, modularity, and commonality of existing locks in the portfolio of RWS. Secondly, methods are sought for to create design specifications for future locks.

The first objective provides insight on how to shape the lock product platform based on the current lock portfolio. In particular, analysis methods are sought:

1. To find groups of similar locks in the lock portfolio of RWS, i.e., groups of locks that share many functions and design characteristics. It is argued that locks within a group can be renovated or replaced using the same set of (semi-)standardized component modules. Hence, the number of groups provides an indication of how many conceptual lock variants one should be able to derive from the lock platform. This number may decrease if RWS decides to no longer build a certain variant in the future or this number may increase if RWS decides to add a new variant.

2. To find modules of lock components within locks based on the system architecture. The basic building blocks of the lock platform are modules of components and their interfaces. System architecture is described as the mapping of a system's functions to the physical components within the system, and the dependencies between those components (Ulrich, 1995). In designing a product platform it is desirable to create modules of components that are as independent as possible (Simpson, 2004).
3. To determine which modules of component and interfaces of components are part of the basic lock structure, and which are part of the optional lock structure. Modules and interfaces that are part of the basic lock structure are the primary candidates for full standardization.
4. To determine which component modules are candidates for full-, semi-, or non-standardization, given the desire of RWS to increase lock reliability and availability (RA), to decrease lock life-cycle-costs (LCC), and to decrease uncertainty in construction costs and time.

The second objective contributes to the actual implementation and usage of the platform. In particular, methods are sought:

5. To create structured and consistent design specifications. RWS outsources the design and construction of locks. To ensure that future locks will meet the predefined standards and interfaces dictated by the lock platform, detailed specifications need to be created for each of the component modules. The consistency of such specifications is essential to ensure the compatibility of the different modules, and to prevent costly and lengthy design iterations.
6. To derive a model of the system architecture directly from design specifications. A visual model of the system architecture helps engineers to increase their understanding of the system, to identify dependencies between components, and to promote communication between engineers (Sosa et al., 2007). For each renovation and replacement project, RWS has to write a public tender. As such, RWS has to work with many different subcontractors. A graphical model of the system architecture aids in the communication and in the transfer of knowledge.

In the next section we summarize the methods we have used and developed to reach the objectives presented above. The Design Structure Matrix is the fundamental modeling concept.

3. Methods

Objective 1 - To identify groups of similar locks a similarity matrix, as presented by Chen (2005), is used. A similarity matrix is a square numerical matrix in which entries have a value of a least 0 and at most 1. A value of 0 at position i, j indicates that elements i and j are 0% similar, while a value of 1 indicates that element i and j are 100% similar. We obtained this matrix by first manually building a characteristic matrix in which the rows are labeled with lock characteristics, such as the type of doors, type of leveling systems, and door-actuators, and the columns are labeled with locks. A non-zero entry within the characteristic matrix at position i, j indicates that lock j possesses characteristic i.

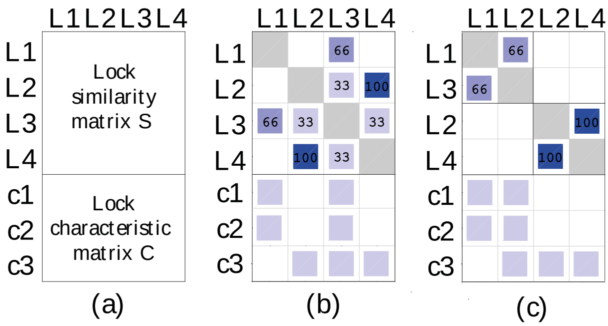


Figure 1: Schematic similarity matrix **S** and characteristic matrix **C**

Subsequently, Jaccard's resemblance coefficient (Jaccard, 1908) is used to calculate the similarity values for all lock pairs based on the characteristics they possess. These values are placed within the similarity matrix. Next, the similarity matrix is pruned to 0.40, i.e., all values below 0.40 are set to zero to increase the sparsity of the matrix as most locks share at least a few characteristics. The pruned matrix is subsequently clustered using the algorithm of Wilschut (2017).

For example, Figure 1a schematically shows characteristic matrix $\boldsymbol{C}$, in which the rows are labelled with characteristics c1, c2, and c3 and the columns are labelled with locks L1, L2, L3, and L4. Figure 1b shows that, for example, lock L1 possesses characteristics c1 and c2 and that lock L3 possesses characteristics c1, c2, and c3. Lock L1 and L3 share two out of the three characteristics they mutually possess, as such they have a similarity of 66% as shown in similarity matrix S. By pruning and clustering similarity matrix $\boldsymbol{S}$, we obtain Figure 1c, in which locks L1 and L3, and locks L2 and L4 are clustered together.

Objectives 2 and 3 – To find modules of components within locks and to determine which modules are common and which are optional, n DSMs are built which are subsequently combined into a ΣDSM $\boldsymbol{F}$ (Gorbea, 2007), schematically depicted in Figure 2. The higher a value within ΣDSM $\boldsymbol{F}$ at position i, j, the more likely that the dependency between component i and component j is present within all locks in the portfolio. $\boldsymbol{F}$ is analyzed using a clustering algorithm to find modules of components that have relatively many mutual dependencies and relatively few external dependencies. For this purpose, we developed a multi-level Markov Clustering algorithm (Wilschut, 2017) that can handle bus structures within the DSM. Modules that have many high valued dependencies are likely to be common. Modules that have many low valued dependencies are not likely to be common.

To fully represent the design variety within the lock portfolio, n should be equal to 127, i.e., the number of locks in the portfolio. However, building 127 DSMs is not feasible within a reasonable amount of time. Therefore, we assume that locks that share many characteristics show little to none variation in system architecture. This enables us to reduce n to the number of lock groups which result from Objective 1, i.e., for each group a single representative lock is chosen.

To ensure that the DSMs, that represent the different groups of locks, can be merged into a single DSM, a single general lock decomposition is made that contains all possible components that a lock may contain. Each representative lock contains a subset of components of the general lock decomposition.

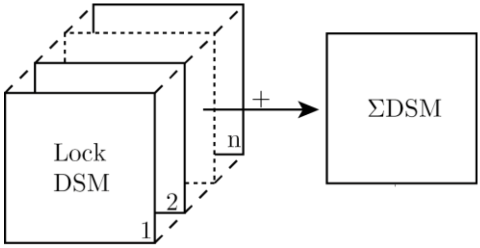


Figure 2. Schematic ∑DSM $\boldsymbol{F}$.

Objective 4 – The method of Brady (2002) is used to determine which component modules are candidates for full-, semi-, or non-standardization, given the desire of RWS to increase lock reliability and availability (RA), to decrease lock life-cycle-costs (LCC), and to decrease uncertainty in construction costs and time. Brady presented a method to identify development risks in the NASA pathfinder architecture. That is, each component in the lock decomposition is given an impact score of 0, 1, 3, or 9 with respect to performance indicators reliability, availability, construction cost, maintenance cost, renovation cost, and life-cycle cost, respectively. The individual component scores are determined using expert interviews as no field-data was readily available. The values are subsequently projected upon ΣDSM $\boldsymbol{F}$ resulting in six projection matrices, i.e., one matrix for each performance measure. For example, reliability projection matrix $\boldsymbol{P}_{\mathrm{R}}(i,j) = \boldsymbol{F}(i,j) \cdot (I_{\mathrm{R},i} + I_{\mathrm{R},j})$, in which $I_{\mathrm{R},i}$ and $I_{\mathrm{R},j}$ are the reliability impact scores for component i and component j, respectively. Thus, the impact score assigned to each component dependency depends on how common that dependency is within the lock variants and the scores assigned to each component.

Objectives 5 and 6 - RWS outsources the design and the construction of locks. To ensure that future locks will meet the predefined standards and interfaces dictated by the lock platform, detailed specifications need to be created for each of the component modules. What is more, the to be renewed locks may have to fulfill additional (function) requirements that may require a change in system architecture. Therefore, we decided to develop a language for the specification of concise and consistent multi-level function- and design specifications from which multi-domain matrix (MDM) models can automatically be generated.

In Wilschut (2018a), we showed that by writing function requirements in terms of goal-functions and transformation-functions following a fixed grammar, a component – function – parameter MDM can be automatically derived. A goal-function denotes the purpose of a component with respect to another component, e.g., to provide power. A transformation-function denotes the internal conversion of flow within a component, e.g., the conversion of power to torque.

The automated generation of MDMs directly from function requirements is the bridge between requirement specification and DSM modeling. Such a bridge is essential in ensuring the compatibility of the various modules of components within the lock product platform. That is, the derived MDMs provide clear insight into the dependencies between the various modules. As such, we continued research into this bridge, which resulted into the Elephant Specification Language (ESL) (Wilschut, 2018c). ESL allows for the creation of function- and design-specifications in terms of needs, requirements, and constraints at multiple granularity levels, following the systems engineering V-model. The system decomposition tree forms the central structure of an ESL specification. The function- and design-needs, requirements, and constraints are specified within the body of component definitions. ESL has a fixed syntax and semantics and supports the formal derivation of dependencies between components, needs, requirements, constraints, variables, and combinations thereof throughout the branches and layers of the system decomposition tree. These dependencies are visualized using DSMs, MDMs, and are analyzed using clustering algorithms.

4. Results

Objective 1 – The clustered similarity matrix revealed that the 127 locks in RWS's lock portfolio can be clustered into seven groups (details presented in Wilschut et al., 2018b). Each group of locks has a distinct combination of characteristics and, therefore, represents a distinct lock variant. Four clusters have a high mutual similarity. Interestingly, most locks that have been built after the year 2000 are a member of the same cluster. As such, this seems the preferred variant in modern-day lock engineering in the Netherlands. Most locks that are due for renewal before 2030 are a member of two distinct clusters. The locks that are due for renewal before 2050 are distributed over four clusters. These results enabled us to categorize the seemingly diverse lock portfolio of RWS into seven lock variants and gain insight in scope of the upcoming renovation and replacement task.

Objectives 2, 3 and 4 – The results of Objective 1 indicate that seven representative locks, i.e., one four each group, can represent the architectural variety in the lock portfolio. In a previous study, Dijkstra (2015) had manually built and analyzed four DSMs of four distinct as-built locks by reviewing design documentation on spatial, information, and energy dependencies. These locks were selected based on expert opinions such that they represent the lock portfolio variety. Not surprisingly, these locks are a member of four different lock groups. Two lock groups do not possess characteristics, different from the other groups, that cause variations in system architecture. These locks primarily differ in geometrical dimensions, which are important from a civil engineering point of view. Therefore, building DSMs for those groups would not yield any additional insight. This left only one group for which an additional DSM had to be built. This DSM has been built during a student project in which the general decomposition of Dijkstra was used as a starting point to allow for easy comparison with Dijkstra's DSMs.

The five DSMs are summed into ΣDSM $\boldsymbol{F}$, which has been subsequently clustered. Component dependencies with a value of at least four are marked as being likely to be common, component dependencies with a value of at least 2 and at most 3 are marked as being semi-likely, and component dependencies with a value of at most one are marked as not likely to be common. Next, the various component impact factors regarding R, A, and LCC are projected upon ΣDSM ***F*** yielding six projection matrices. Each of these matrices is separately discussed in Wilschut et al. (2018b).

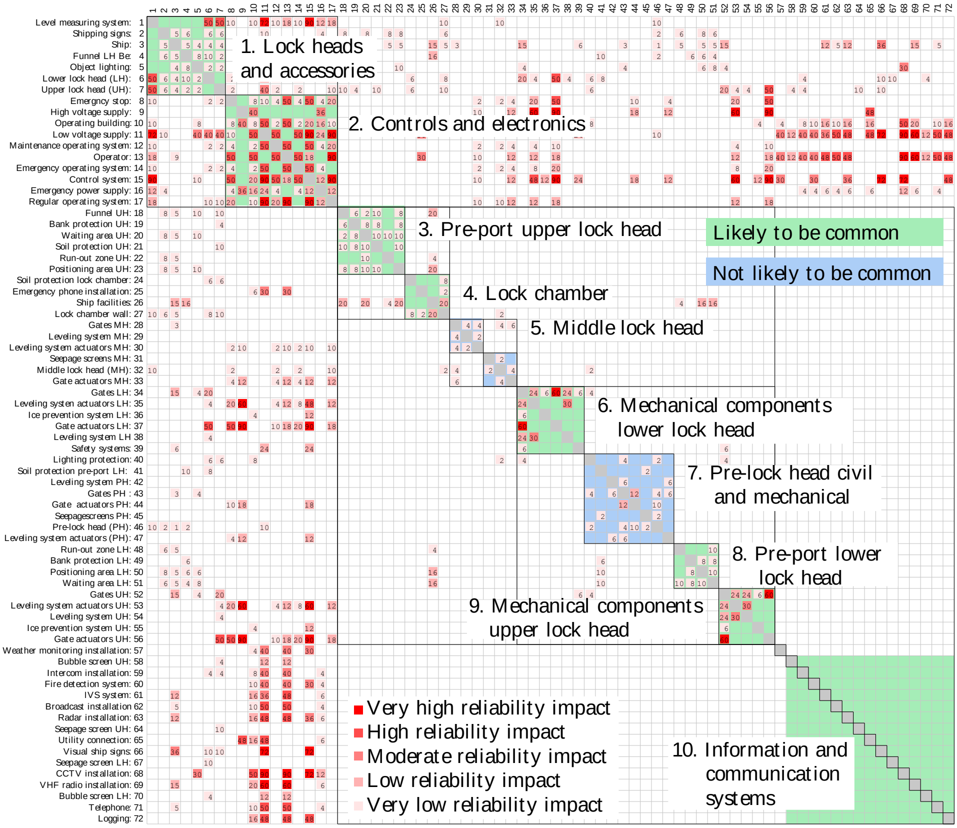


Figure 3. Reliability projection matrix $\mathbf{P}_R$.

Figure 3 combines the results of ΣDSM ***F*** with the result of reliability projection matrix $\boldsymbol{P}_R$. That is, clusters are shaded green if they contain primarily component dependencies that are marked as likely to be common and clusters are shaded blue if they primarily contain component dependencies that are marked as not likely to be common. The dependency values indicate the reliability impact score.

Note that eight out of the ten component clusters are likely to be common. However, internally these contain component dependencies that are semi-likely to be common (not visible here). These variations are often due to variations in working principle or in embodiment of components, and not due to variations in desired functionality. Thus, the next step in RWS' standardization efforts should focus on selecting preferred working principles and embodiment of components to reduce architectural design variety in their lock portfolio.

The reliability dependency impact scores are particularly useful to draw conclusions on a cluster level. Figure 3, for example, clearly shows that Clusters 2, 6, and 9 have the highest impact on the reliability of the lock portfolio. As such, in selecting preferred working principles and embodiment of components in these clusters, RWS should carefully evaluate the reliability of each option.

```
goal-requirement
  gr-wf-01: lock-complex-x must regulate water-out-flow-1 to water-way-1 ...
    with subclauses
      c-02: reliability must be at least 'TBD' [%/year]
      c-03: availability must be at least 'TBD' [%/year]
      c-04: upper-flow-range must be at least 'TBD' [m3/s]
      c-05: lower-flow-range must be at most 'TBD' [m3/s]
    end

  gr-wf-02: lock-complex-x must regulate water-out-flow-2 into water-way-2 ...
    with subclauses
      c-02: reliability must be at least 'TBD' [%/year]
      c-03: availability must be at least 'TBD' [%/year]
      c-04: upper-flow-range must be at least 'TBD' [m3/s]
      c-05: lower-flow-range must be at most 'TBD' [m3/s]
    end
```

Figure 4. An example ESL specification.

Overall, the six projection matrices enabled us to determine which component modules should be targeted if one wants to improve a certain performance indicator and which performance indicators are most important while selecting preferred working principles and embodiment of components within modules.

Objective 5 and 6 – A dedicated language, referred to as Elephant Specification Language (ESL), has been developed to support the creation of concise and unambiguous function- and design-specifications for (semi-) standardized modules of components (Wilschut et al., 2018c). In Wilschut et al. (2018d), we present the first proof of principle of ESL in a pilot study concerning a lock renovation project. In this pilot study, we converted natural language requirement statements, such as:

> “*SYS-0194: The navigation lock must retain high water without any unacceptable leakage flow*” (Nieman, 2016, translated from Dutch).

into ESL statements, as shown in Figure 4. Each goal-requirement consists of a main-clause, stating the function that must be fulfilled, and zero or more sub-clauses that state additional conditions that must be fulfilled.

ESL distinguishes between requirements and constraints. Requirements denote what is desired, while constraints denote limitations on what is desired. We used this feature to visualize the impact of a renovation project. For example, Figure 5 shows the component DSM at decomposition level 2 that has been automatically generated from an ESL specification, in which the functions and design of components that are due for renewal are specified in terms of requirements and the functions and design of components that are not due for renewal are specified in terms of constraints. This DSM is part of a larger component – goal-function MDM presented in Wilschut (2018d, 2018e). In Figure 5, the component DSM shows the various types of dependencies between components and several clusters. All dependencies and components that are marked with a red circle are affected by the renovation, i.e., those dependencies are derived from requirements. For example, all electrical-energy-flow dependencies are marked as the locks power-supply is due for replacement.

The results show that ESL has sufficient expressiveness and flexibility to capture the content of natural language requirement documents. Additionally, the generated DSMs and MDMs provide direct insight into the system architecture, and in particular, into which component interfaces area affected by the renovation. This enables engineers to quickly focus their efforts in designing the replacement parts.

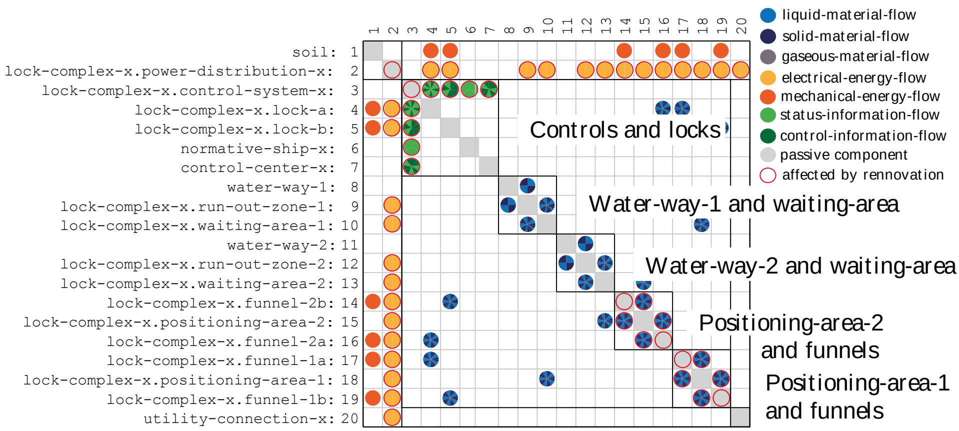


Figure 5. Generated component DSM.

5. Closing remarks

The presented DSM based methods enabled us: (a) to bring structure to the seemingly diverse lock portfolio of RWS; (b) to find the similarity, modularity, and commonality of locks; and (c) to identify component modules with a significant impact on RA and LCC. That is, the clustered similarity matrix revealed that the 127 locks in the portfolio can be grouped into seven groups which possess a distinct combination of characteristics. As such, the future lock platform should support the development of seven locks variants (if one decides to maintain all variants). The comparison of the system architecture of five locks that are part of different groups using a ΣDSM, revealed that most variety in system architecture designs results from differences in working principle and embodiment of components, not from differences in provided functionality. As such, RWS can reduce the design variety by selecting preferred (standard) working principles and embodiments for components. The projection matrices revealed which component modules should be targeted if one wants to improve a certain performance indicator, and thus, are candidates for full standardization.

The developed specification language ESL, enables one to concisely and unambiguously create function- and design-specifications for modules of components in the lock platform. Additionally, when ESL is used to describe an existing lock, one can quickly gain insight into the system architecture and identify those component interfaces that are affected by, for example, the implementation of a new (standardized) module.

Acknowledgements

We would like to thank Rijkswaterstaat, part of the Dutch Ministry of Infrastructure and Environment, which has generously provided funding for this research. In particular we would like to thank, Maria Angenent, Erik-Jan Houwing and Robert de Roos for their enthusiastic involvement in this project.

References

Alblas A. A. and Wortmann. J. C., 2012 Impact of product platforms on lean production systems: evidence from industrial machinery manufacturing. *International Journal of Technology Management*, 57(1):110-131.

Alblas, A. A. and Wortmann J. C., 2014. Function-technology platforms improve efficiency in high-tech equipment manufacturing: A case study in complex products and systems (CoPS*). International Journal of Operations & Production Management*, 34(4):447-476.

Brady, T. K., 2002. Utilization of dependency structure matrix analysis to assess complex project designs. *Proceedings of ASME 2002 IDETC/CIE Conference*, Montreal, Canada, 231-240.

Chen, L. and Li, S., 2005. Analysis of Decomposability and Complexity for Design Problems in the Context of Decomposition. *Journal of Mechanical Design*, 127(4).

Dijkstra, M., 2015. Reliability and availability risk identification within a lock's dependency structure. M.S. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.

Gorbea, C., 2007. Fricke, E., and Lindemann, U., 2007. Pre-selection of hybrid electric vehicle architectures during the initial design phase. *DSM 2007: Proceedings of the 9th International DSM Conference*, Munich, Germany.

Jaccard, P., 1908. Nouvelles recherches sur la distribution orale. *Bulletin de la Societe vaudoise des sciences naturelles*, 44.

Jiao, J., Tseng, M. M., 1999. A methodology of developing product family architecture for mass customization*. Journal of Intelligent Manufacturing*, 10(1):3-20.

Jiao, J., Simpson, T. W., Siddique, Z.. 2007. Product family design and platform-based product development: a state-of-the-art review. *Journal of Intelligent Manufacturing*, 18(1):5-29

Meyer, M., Lehnerd, A. P., 1997. *The power of product platform - building value and cost leadership*. Free Press, New York, USA.

Nieman, W., 2016. Basisspecificatie Schutsluis. Technical report, Ministry of Infrastructure and Water Management, The Netherlands. In Dutch.

Simpson, T. W., 2004, Product platform design and customization: Status and promise. *AI EDAM*, 18(01):3-20

Sosa, M. E., S. D. Eppinger, S. D., Rowles, C. M., 2007. Are your engineers talking to one another when they should*? Harvard Business Review*, 85(11):133, 2007.

Ulrich, K., 1995. The role of product architecture in the manufacturing firm. *Research Policy*, 24(3):419-440.

Wilschut, T., Etman, L. F. P., Rooda, J. E., Adan, I. J. B. F., 2017. Multi-level flow-based Markov clustering for design structure matrices. *Journal of Mechanical Design*, 139(12):1-10.

Wilschut, T., Etman, L. F. P., Rooda, J. E., Vogel. J. A., 2018a. Automated generation of a function-component-parameter multi-domain matrix from textual function specifications. *Research in Engineering Design*, doi: https://doi.org/10.1007/s00163-018-0284-9.

Wilschut, T., Etman, L.F.P., Rooda, J.E., Vogel. J.A., 2018b. Similarity, modularity, and commonality analysis of navigation locks in The Netherlands. Working Paper

Wilschut, T., Etman, L.F.P., Rooda, J.E., Vogel. J.A., 2018c. A language for multi-level system specification and automated derivation of component, function, and variable dependencies. Working Paper.

Wilschut, T., Etman, L. F. P., Rooda, J. E., Vogel. J. A., 2018d. Multi-level function specification and architecture analysis using ESL: a lock renovation project pilot study. *In Proceedings of the ASME 2018 IDETC/CIE.* Quebec City, Quebec, Candada.

Contact: T Wilschut, Eindhoven University of Technology, Department of Mechanical Engineering, PO BOX 513, Eindhoven, The Netherlands, t.wilschut@tue.nl

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Defining system boundaries in change propagation analysis: A diesel engine case study

Edwin C.Y. Koh[1], Nicholas H.M. Caldwell[2], P. John Clarkson[3]

[1]Engineering Design and Innovation Centre, National University of Singapore
[2]School of Science, Technology and Engineering, University of Suffolk
[3]Engineering Design Centre, University of Cambridge

Abstract: This paper explores how change propagation analysis can be affected by the way system boundaries are defined. This is an important issue as engineering change can in reality propagate out of the system modelled and back through components that were not considered. The work builds on a diesel engine case study to examine the difference in analysis results generated based on a full system model (i.e. entire engine) and those generated based on a set of partial system models (e.g. sub-assemblies). It was found that partial system models with boundaries defined by physical sub-assemblies can produce analysis results that are highly correlated with the one produced using a full system model. It was also revealed that modelling more components (i.e. a more complete system model) does not necessarily increase the level of correlation. The findings can be used to support system boundary decisions in change propagation analysis.

Keywords: Change propagation, Changeability, System boundaries

1 Introduction

It is widely accepted that complex engineering systems are often designed through modifications of existing ones (Giffin et al., 2009; Shankar et al., 2012; Fernandes et al., 2015). Such an approach can facilitate the reuse of components and knowledge from previous designs. However, it is documented that changes initially perceived as simple can sometimes propagate undesirably, resulting in costly delays (Eckert et al., 2004; Duran-Novoa et al., 2018). Hence, modelling approaches have been developed to support the management of engineering change propagation in design projects (Siddiqi et al., 2011; Koh et al., 2012; Maier et al., 2014; Lee and Hong, 2017; Ma et al., 2017) and across life cycle of products and systems (Vianello and Ahmed-Kristensen, 2012; Hu and Cardin, 2015; Luo, 2015).

While efforts have been made to discuss how change analysis results can be affected by model granularity (Maier et al., 2017) and the types of change data used (Koh, 2017), few studies discuss how change analysis results can be affected by the way system boundaries are defined. System boundary decisions are especially important in change propagation analysis as engineering change can in reality propagate out of the system modelled and back through components that were not considered (See Figure 1). Yet, the issue of system boundaries in change propagation analysis is often overlooked as the components to be modelled are usually pre-defined based on the needs and constraints of the analysis. For example, engineering change analysis may be conducted on an engine short block, an entire engine, or an entire truck, depending on whether the analysis is for a supplier of engine parts, a producer of engines, or a truck manufacturer. In addition, information on components designed by other stakeholders (e.g. suppliers, collaborators) may be unavailable for modelling. Even if all system components are designed within the same organisation, the resources required to model the full system can be a challenge as well. Therefore, it is not uncommon to see change propagation analysis conducted based on partial system models rather than full system models (e.g. analysing an engine instead of a full truck). This raises several questions: Can partial system models produce valid change propagation analysis results? Will the validity of change propagation analysis improve when more system components are modelled? How might the validity of change propagation analysis be affected if the boundaries for the partial system to be modelled were arbitrarily determined due to a lack of information? To address these questions, this paper presents an exploratory study that discusses how change propagation analysis results can be affected by the way system boundaries are defined.

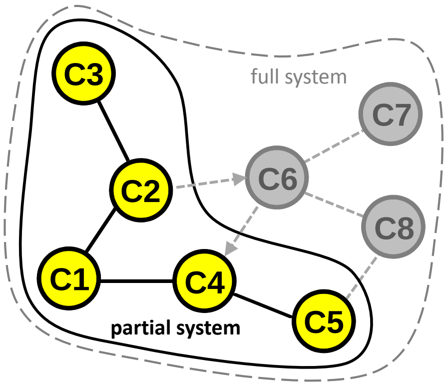


Figure 1: An example of a partial system model missing a change propagation path (C2 to C6 to C4) during analysis

2 Research Approach

To ensure that realistic change analysis results can be generated and analysed, the modelling data used to create system models in this work were extracted from the industry case study published in (Koh et al., 2013). For consistency, the change analysis method used to analyse the data is also adapted from (Koh et al., 2013). The goal is to examine whether change analysis carried out using partial system models can produce results that are as valid as the one produced using a full system model. In this paper, results produced by partial system models are considered to be as valid as the one produced using the full system model if they are found to be correlated, resulting in similar design decisions. The following sections provide details on the modelling data used (Section 2.1), the change analysis made (Section 2.2), and the correlation study carried out in this work (Section 2.3).

2.1 Modelling data

The full data set used in this work describes a heavy-duty diesel engine comprising of 32 components. Figure 2 shows an excerpt of the modelling data presented in the form of

Design Structure Matrices (DSMs). Each row and column heading represents a particular component. For example, 'CB' refers to the 'Cylinder Block' and 'P' refers to the 'Piston' (not all component names are disclosed for confidentiality reasons). The DSM on the left of Figure 2 describes the change likelihood of each component. For instance, the diagonal cells in the DSM describe the likelihood of changing a given component due to exogenous factors, such as a change in design requirements. The off-diagonal cells describe the likelihood of changing a given component (indicated by the row heading) due to changes in another component (indicated by the column heading). Note that the entries were based on a quantitative {0; 0.25; 0.5; 0.75} scale that represents 'Nil', 'Low', 'Medium', and 'High' strength levels, respectively. For example, the likelihood of changing the Cylinder Block (CB) due to exogenous factors is 'Medium' (i.e. Entry for Column 2 and Row 2 is '0.5'), and the likelihood of changing the Cylinder Block (CB) due to changes in the Piston (P) is also 'Medium' (i.e. Entry for Column 3 and Row 2 is '0.5').

Change Likelihood		Initiating					
		CH	CB	P	CS	EA	…
Affected	CH	0.25	0.50	0.50	0.00	0.50	…
	CB	0.75	0.50	0.50	0.50	0.50	…
	P	0.25	0.25	0.25	0.00	0.00	…
	CS	0.00	0.50	0.50	0.25	0.00	…
	EA	0.75	0.50	0.00	0.00	0.25	…
	…	…	…	…	…	…	…

Change Impact		Initiating					
		CH	CB	P	CS	EA	…
Affected	CH	0.75	0.75	0.50	0.00	0.50	…
	CB	0.75	0.75	0.75	0.50	0.50	…
	P	0.25	0.50	0.75	0.00	0.00	…
	CS	0.00	0.75	0.25	0.75	0.00	…
	EA	0.25	0.25	0.00	0.00	0.25	…
	…	…	…	…	…	…	…

Figure 2: An excerpt of the modelling data used (adapted: Koh et al., 2013)

The DSM on the right of Figure 2 describes the change impact of each component. The diagonal cells describe the average change impact (based on redesign cost) of changing a given component while the off-diagonal cells describe the average proportion of redesign work required if changes propagate from a given component (indicated by the column heading) to another component (indicated by the row heading). For example, the impact of changing the Cylinder Block (CB) in terms of redesign cost is 'High' (i.e. Entry for Column 2 and Row 2 is '0.75'). The impact of changing the Cylinder Block (CB) due to changes in the Piston (P) is also 'High' in terms of the average proportion of redesign work required (i.e. Entry for Column 3 and Row 2 is '0.75').

Based on the full data set of the entire diesel engine, the 32 engine components were later sorted based on how the diesel engine was divided into sub-assemblies. For example, each engine component has a unique four-digit serial number with the first two digits indicating the sub-assembly that it belongs to. By sorting all the serial numbers, it was found that the engine consists of six sub-assemblies with distinct components in each sub-assembly (see Table 1). Sub-assembly A has 18 components. It is the main sub-assembly and forms the 'Long Block' of the engine. Sub-assembly B has 6 components. Sub-assembly C and D have 3 components each. Sub-assembly E and F have 1 component each. Subsequently, by organising the sub-assemblies into groups, 6 sets of system model were created as shown in Table 2 (see SM1 to SM6). SM1 is the full system model of the diesel engine and comprises all the sub-assemblies. It was created as the reference model to be compared with. SM2 to SM6 were created based on boundaries defined by the sub-assemblies and represent scenarios where only parts of a full system are modelled. For instance, SM2 is a partial system model of the diesel engine consisting of just the main sub-assembly (i.e. Sub-assembly A, the 'Long Block'). SM3, SM4, and SM5 are partial system models created by adding more sub-assemblies to SM2, with the purpose of exploring the effect of modelling more components (i.e. towards a more complete system model compared to SM2). SM6 is a partial system model that excludes only the main sub-assembly (i.e. Sub-assembly A) and was created to better understand the influence of the main sub-assembly in this work.

Table 1. A breakdown of diesel engine sub-assemblies

Sub-assembly reference	Number of components
A	18
B	6
C	3
D	3
E	1
F	1
Full system	32

Table 2. A breakdown of system models to be tested

System model reference	Description
SM1	Full System
SM2	A
SM3	A + B
SM4	A + C
SM5	A + D + E + F
SM6	B + C + D + E + F
SM7	Random 1
SM8	Random 2
SM9	Random 3

As mentioned, SM2 to SM6 were created based on boundaries defined by physical sub-assemblies and identified through serial numbers. However, in practice, there might be cases where it may not be easy to identify the boundaries for the partial system to be modelled. A hypothetical example is when a junior engineer tries to analyse the 'Long Block' of the engine, but does not know what components to include in the model. Hence, in an attempt to explore the scenario where the boundaries for the partial system to be modelled were arbitrarily determined, 3 further sets of partial system model were created by randomly removing 50% of the engine components from the full system model (see SM7 to SM9 in Table 2). The removed components were identified by using Microsoft Excel to generate a random decimal number next to each component and subsequently ranking the components based on the random decimal numbers generated.

Components that were ranked in the top 50% were removed while those in the bottom 50% were selected to form a partial system model. The process was repeated 3 times to create the 3 randomly generated models – SM7, SM8, and SM9. A breakdown of the number of sub-assembly components in these randomly generated models is presented in Table 3.

Table 3. A breakdown of the randomly generated models

Sub-assembly reference	Number of sub-assembly components in model SM7	SM8	SM9
A	8	10	9
B	3	3	3
C	3	2	0
D	1	1	3
E	0	0	0
F	1	0	1

2.2 Change analysis

The change analysis method documented in (Koh et al., 2013) is adapted in this study to process the system models described in Section 2.1. The method is a matrix-based technique that systematically examines the changeability of system components by considering exogenous changes (e.g. new customer requirements) and endogenous changes (e.g. change propagation between components). It extends the conventional Change Prediction Method introduced by (Clarkson et al., 2004) by considering the reachability of change propagation in its algorithms, which effectively limits the maximum length of change propagation paths to be examined by taking into account resource constraints for changes to propagate further. The analysis results derived from the method can be used to rank system components in terms of change risk and support design decisions, such as the planning of modularisation efforts based on the rankings produced (Koh et al., 2015).

Figure 3 shows how the data presented in Figure 2 were processed. The first step was to revise the change propagation likelihood between components using Equation 1 to 4 expressed as follows:

$$L_{k,j}^* = L_j \times L_{k,j} \qquad (1)$$

$$L_{k,j} = 1 - \prod_{z \in Z} \left[1 - \left(l_z \times \alpha_z\right)\right] \qquad (2)$$

$$l_z = \left(l_{k,k-1} \times l_{k-1,k-2} \times \ldots \times l_{j+1,j}\right) \qquad (3)$$

$$\alpha_z = \left(\alpha_{k,k-1} \times \alpha_{k-1,k-2} \times \ldots \times \alpha_{j+1,j}\right) \qquad (4)$$

$L_{k,j}$* represents the revised change propagation likelihood from component '*j*' to '*k*' where L_j represents the likelihood of changing component '*j*' due to exogenous factors and $L_{k,j}$ represents the combined (direct and indirect) change propagation likelihood from component '*j*' to '*k*', with '*j*' representing the change initiating component and '*k*' representing the last component in change propagation path '*z*' and '*Z*' representing the entire set of change propagation paths from component '*j*' to '*k*'. l_z represents the change propagation likelihood for a particular path ‘z’ where the individual $l_{k,k-1}$ represents the direct change propagation likelihood between successive components along path ‘z’. α_z represents the change propagation reachability for a particular path ‘z’ where the individual $\alpha_{k,k-1}$ represents the change propagation reachability between successive components along path ‘z’.

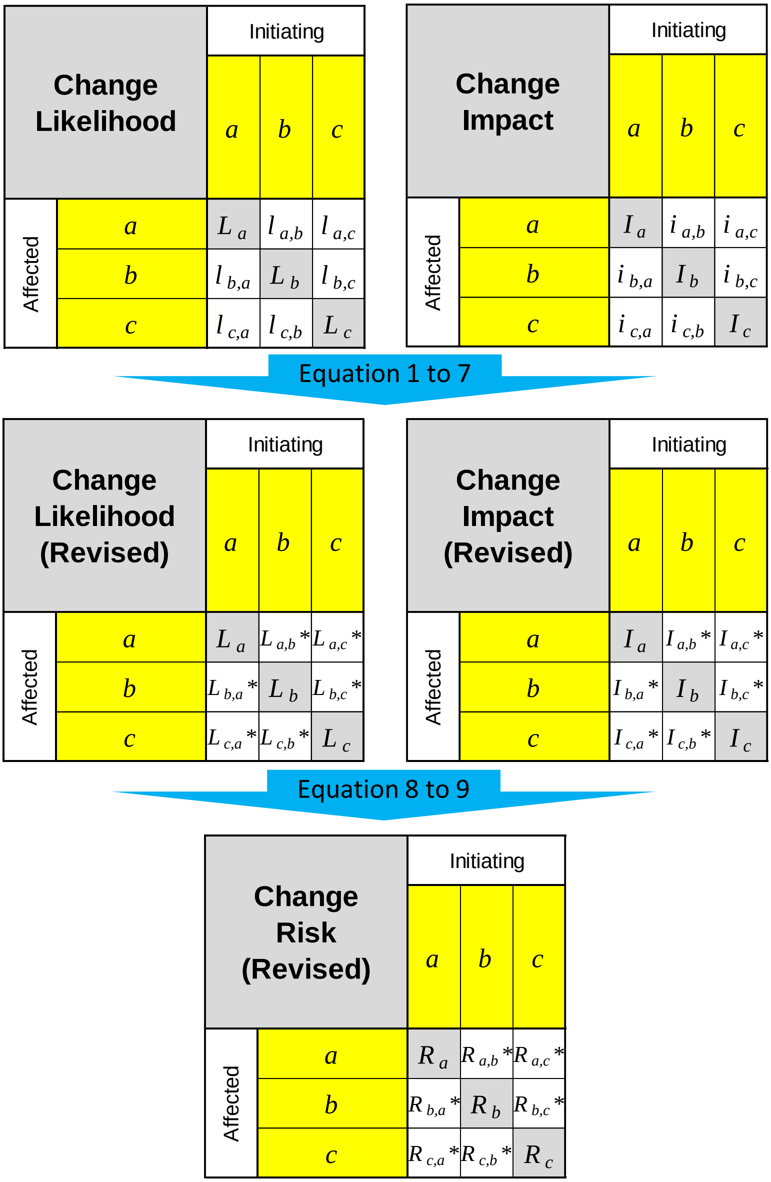


Figure 3: Using change likelihood and impact to compute change risk

Subsequently, the change propagation impact between components were revised using Equation 5 to 7. The equations are expressed as follows:

$$I_{k,j}^{*} = I_{k,j} \times I_k \quad (5)$$

$$I_{k,j} = \frac{1 - \prod_{z \in Z} \left[1 - \left(i_z \times \alpha_z\right)\right]}{L_{k,j}} \quad (6)$$

$$i_z = \left(i_{k,k-1} \times l_z\right) \quad (7)$$

$I_{k,j}$* represents the revised change propagation impact from component '*j*' to '*k*' where I_k represents the impact of changing component '*k*' in terms of the average cost of redesigning component '*k*' and $I_{k,j}$ represents the combined change propagation impact in terms of the proportion of redesign work through the change propagation paths. i_z represents the change propagation impact for a particular path '*z*' where $i_{k,k-1}$ represents the direct change propagation impact on the last component caused by the penultimate component in path '*z*'.

After the revised change propagation likelihood and impact were computed, the revised change propagation risk between components (endogenous change risk), the change risk of each component due to exogenous factors (exogenous change risk), and the overall change risk of each component (endogenous and exogenous change risk) were calculated using Equation 8, 9, and 10, respectively. The equations are expressed as follows:

$$R_{k,j}^{*} = L_{k,j}^{*} \times I_{k,j}^{*} \quad (8)$$

$$R_k = L_k \times I_k \quad (9)$$

$$CR_k = \frac{R_k + \sum R_{k,j}^{*}}{n} \quad (10)$$

$R_{k,j}$* represents the revised change propagation risk from component '*j*' to '*k*'. R_k represents the change risk of component '*k*' due to exogenous factors. CR_k represents the overall change risk of component '*k*' due to exogenous factors and change propagation from all other components in the system (see Figure 4). *n* is the number of components in the system. The above process was repeated for all the system models shown in Table 2.

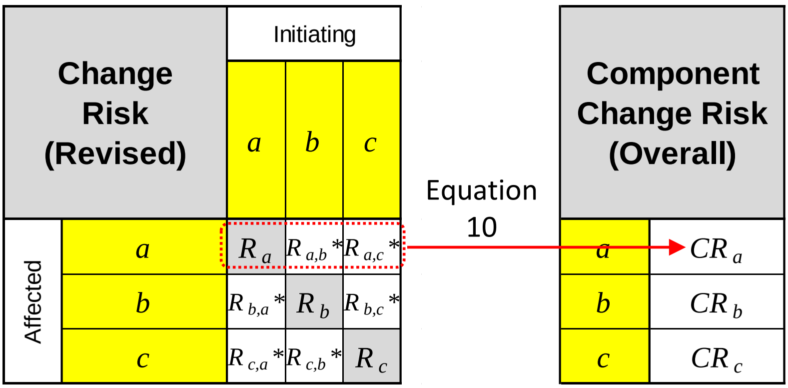


Figure 4: Computing overall component change risk

2.3 Correlation study

Table 4 shows the ranking of components according to their overall change risk based on the change analysis carried out using the full system model (SM1). The table reveals that the Cylinder Block (CB) was ranked 1st as it has the highest overall change risk with a normalised *CR* value of '1.00'. The Piston (P) has a normalised *CR* value of '0.34' and is ranked 12th. As discussed previously, the ranking can be used to support design decisions such as the planning of modularisation efforts. For instance, components with higher ranking have greater change risk and should be assigned a higher priority to be made more modular (Koh et al., 2015). Therefore, with reference to Table 4, the Cylinder Block (CB) should be considered for modularisation ahead of the Piston (P).

Table 4. Ranking of component overall change risk based on the full system model (SM1)

Component	Normalised CR	Ranking
CB	1.00	1
CH	0.77	2
…	…	…
P	0.34	12
…	…	…
EA	0.02	31
GR	0.00	32

Given that the ranking of components may vary if the change analysis was conducted using a different system model, a Spearman's rank correlation study was carried out to examine whether the rankings produced by the partial system models (i.e. SM2 to SM9) are correlated with the one produced using the full system model (i.e. SM1, reference model). As partial system models have fewer components compared to the full system model, components that do not appear on both sets of ranking during correlation study were removed to create ranking sets with the same number of components. For example, in the correlation study between SM1 and SM2, components that are not in Sub-assembly A were removed from the full system ranking to create two sets of ranking with exactly 18 components, ranking from 1st to 18th (see Table 5). A given partial system model is considered to have produced change analysis results as valid as the one produced using the full system model if the component rankings were found to be correlated (i.e. both models produced rankings that will lead to similar design decisions). Results of the correlation study is presented in Section 3.

Table 5. Ranking of Sub-assembly A components based on SM1 and SM2

	SM1		SM2	
Sub-assembly A Components	Normalised CR	Ranking*	Normalised CR	Ranking
CB	1.00	1	1.00	1
CH	0.77	2	0.66	2
…	…	…	…	…
P	0.34	7	0.35	7
…	…	…	…	…
EA	0.02	17	0.00	18

*Only components from Sub-assembly A are included

3 Results

Table 6 shows the results of the Spearman's rank correlation analysis carried out in this work. It can be seen that the Spearman's coefficients range from '0.94' to '0.98' with partial system models that were defined based on physical sub-assemblies (i.e. SM2 to SM6, see Table 2). However, the Spearman's coefficients for randomly generated system models (i.e. SM7 to SM9, see Table 2) are lower and range from '0.65' to '0.81'.

Table 6. Spearman's rank correlation coefficient for SM2 to SM9

	SM2	SM3	SM4	SM5	SM6	SM7	SM8	SM9
Spearman's coefficient* (rank correlation with SM1)	0.96	0.94	0.96	0.98	0.97	0.81	0.78	0.65
Components modelled (out of 32 in SM1)	18	24	21	23	14	16	16	16
Components modelled (out of 100% in SM1)	56%	75%	66%	72%	44%	50%	50%	50%

*P-value less than 0.01

The results shown in Table 6 reveal that partial system models with system boundaries defined based on physical sub-assemblies (SM2 to SM6) can produce change analysis results that are highly correlated with the one produced using the full system model (SM1). However, the validity of change analysis carried out using randomly generated partial system models (SM7 to SM9) is questionable as the correlation can go as low as '0.65' (see SM9 in Table 6). Although it is unlikely that one would knowingly carry out change analysis on randomly generated partial systems, the result suggests that partial system models with poorly defined system boundaries can affect the validity of change analysis.

The results also show that the level of correlation is insensitive to the number of components modelled in the full system. For example, SM3 has 24 components (75% of the full system) and is the largest partial system model. However, the Spearman's coefficient for SM3 is lower than the other partial system models that were not randomly generated (i.e. SM2, SM4 to SM6). In fact, even though SM6 is the smallest with 14 components (44% of the full system), it produced a Spearman's coefficient that is greater than SM2, SM3, and SM4. This suggests that modelling more components (i.e. a more complete system) does not necessarily result in higher validity.

4 Conclusions

The propagation of engineering change is a recognised phenomenon in design. A common challenge in engineering change propagation analysis is to define the system boundaries to be examined. Based on the analyses conducted in this work, it was revealed that partial system models with system boundaries defined based on physical sub-assemblies can produce change analysis results that are highly correlated with the one produced using a full system model. It was also found that modelling more components (i.e. a more complete system) does not necessarily increase the level of correlation. Future work will examine a wider range of engineering systems with different change analysis methods.

References

Clarkson, P.J., Simons, C., Eckert, C.M., 2004. Predicting change propagation in complex design. Journal of Mechanical Design 126 (5), 765-797.

Duran-Novoa, R., Weigl, J.D., Henz, M., Koh, E.C.Y., 2018. Designing in young organisations: engineering change propagation in a university design project. Research in Engineering Design, Published online, https://doi.org/10.1007/s00163-018-0292-9, 1-18.

Eckert, C.M., Clarkson, P.J., Zanker, W., 2004. Change and customisation in complex engineering domains. Research in Engineering Design 15 (1), 1-21.

Fernandes, J., Henriques, E., Silva, A., Moss, M.A., 2015. Requirements change in complex technical systems: an empirical study of root causes. Research in Engineering Design 26 (1), 37-55.

Giffin, M., de Weck, O., Buonova, G., Keller, R., Eckert, C.M. Clarkson, P.J., 2009. Change propagation analysis in complex technical systems. Journal of Mechanical Design 131 (8), 081001, 1-14.

Hu, J., Cardin, M.-A., 2015. Generating flexibility in the design of engineering systems to enable better sustainability and lifecycle performance. Research in Engineering Design 26 (2), 121-143.

Koh, E.C.Y., 2017. A study on the requirements to support the accurate prediction of engineering change propagation. Systems Engineering 20 (2), 147-157.

Koh, E.C.Y., Caldwell, N.H.M., Clarkson, P.J., 2012. A method to assess the effects of engineering change propagation. Research in Engineering Design 23 (4), 329-351.

Koh, E.C.Y., Caldwell, N.H.M., Clarkson, P.J., 2013. A technique to assess the changeability of complex engineering systems. Journal of Engineering Design 24 (7), 477-498.

Koh, E.C.Y., Forg, A., Kreimeyer, M., Lienkamp, M., 2015. Using engineering change forecast to prioritise component modularisation. Research in Engineering Design 26 (4), 337-353.

Lee, J., Hong, Y.S., 2017. Bayesian network approach to change propagation analysis. Research in Engineering Design. 28 (4), 437-455.

Luo, J., 2015. A simulation-based method to evaluate the impact of product architecture on product evolvability. Research in Engineering Design 26 (4), 355-371.

Ma, S., Jiang, Z., Liu, W., 2017. Multi-variation propagation prediction based on multi-agent system for complex mechanical product design. Concurrent Engineering 25 (4), 316-330.

Maier, J.F., Eckert, C.M., Clarkson, P.J., 2017. Model granularity in engineering design – concepts and framework. Design Science 3 (1), 1-29.

Maier, J.F., Wynn, D.C., Biedermann, W., Udo Lindemann, U., Clarkson, P.J., 2014. Simulating progressive iteration, rework and change propagation to prioritise design tasks. Research in Engineering Design 25 (4), 283-307.

Shankar, P., Morkos, B., Summers, J.D., 2012. Reasons for change propagation: a case study in an automotive OEM. Research in Engineering Design 23 (4), 291-303.

Siddiqi, A., Bounova, G., de Weck, O., Keller, R., Robinson, B., 2011. A posteriori design change analysis for complex engineering projects. Journal of Mechanical Design 133 (10), 101005, 1-11.

Vianello, G., Ahmed-Kristensen, S., 2012. A comparative study of changes across the lifecycle of complex products in a variant and a customised industry. Journal of Engineering Design 23 (2), 99-117.

Contact: Dr Edwin Koh, National University of Singapore, Faculty of Engineering, Engineering Design and Innovation Centre. Email: edwin.koh@nus.edu.sg

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

A hunt for the hidden reasons behind a product architecture

David Williamsson[1], Ulf Sellgren[2], Anders Söderberg[2]

[1]Scania CV AB & KTH Royal Institute of Technology
[2]KTH Royal Institute of Technology

Abstract: A modular product architecture is a strategic means to deliver external variety and internal commonality. In this paper, a heavy duty modular gearbox architecture is represented and analyzed. In focus is re-engineering of hidden technical complexity and business strategy concerns behind an existing product architecture. The architecture of the investigated gearbox is represented and analyzed with a *Product Architecture* DSM and the *Integrated Modularization Method* (IMM). Furthermore, a *Cluster Match Matrix* (CMM) is proposed as a means to compare multiple clustering results. The case study indicates that the IMM methodology and CMM can be used for analyzing and finding the explicit and/or implicit reason for a targeted existing product architecture.

Keywords: Product Architecting, Integrated Modularization, DSM, MFD, IMM.

1 Introduction

Ulrich (1995) defined product architecture as “the scheme by which the function of a product is allocated to physical components”, or more formally as: (1) the arrangement of *functional elements,* (2) the mapping from *functional elements* to *physical components* (also referred as technical solutions) and (3) the specification of the *interfaces* among interacting system components.

The architecture of a product may be categorized based on the type of mapping between functional elements and physical components. If there is a one-to-one mapping between functional elements and physical components, the design is said to be uncoupled, while it is said to be coupled if the mapping is complex. In 2005, Hölttä-Otto defined these two types of architectures as being modular (uncoupled) and integral (coupled). Thus, a module is a configuration of highly interconnected system elements with few interrelations with components outside of the module (Ulrich, 1995). This implies that the architecture of a module may very well be integral. A common definition is that a *module* is a functional building block, with well-defined and standardized interfaces between modules, and that it should be chosen for company specific reasons, i.e. support a company specific business strategy (Erixon 1998). A *module variant* is a physical incarnation of a module with a specific performance level or appearance. A module may therefore have multiple module variants, which may be configured in multiple ways in order to satisfy different customer requirements. Thus, a *modular system* can be defined as the collection of module variants by which all the required end products can be built (Börjesson, 2014).

Hölttä-Otto (2005) presented the following three main approaches for modularizing a product; *Heuristics*, *Modular Function Deployment* (MFD) and *Design Structure Matrix* (DSM). Heuristics is based on an analysis of the pattern of flow of matter, energy, and information between function blocks, see e.g. (Erixon, 1998). MFD (Erixon, 1998) (Ericsson and Erixon, 1999) is a five-step method for translating customer requirements into a modular architecture, while considering the strategic company specific objectives, represented by twelve predefined generic *Module Drivers* (MD:s) that should reflect the strategic objectives of the company, e.g. modules can reduce capital needs and bring economies in parts sourcing (Baldwin and Clark, 2000), (Ulrich and Tung 1991). In the MFD methodology, the MD:s are represented by a *Module Indication Matrix* (MIM), which is an interdomain matrix that relates the physical function carriers, i.e. the components, and the twelve MD:s. The main focus of DSM-based modularization approaches is to minimize technical complexity by clustering the component-DSM in a way that minimize the technical interactions between clusters of components, i.e. complex interactions are grouped within clusters. A cluster is a module candidate.

There are two main categories of relations or interactions that are important to consider when representing the product architecture, i.e. hierarchical (vertical) and lateral (horizontal). Hierarchical relations are used when modeling a breakdown of a product into subsystems, modules and components etc., e.g. a *product breakdown structure* (PBS), also referred to as a *product structure*. Lateral relations describe how the elements in the product architecture interact, at a given level of decomposition. Hence, different types of relations can be represented in the DSM. Pimmler & Eppinger (1994) proposed four generic interaction types to represent the lateral relations between the technical solutions or functions in a Product Architecture DSM. These are *spatial relations* and *flow of matter*, *information* and *energy*. Some relations may be more important than others. *Relation weights*, also known as *interaction strengths,* are therefore used to represent their relative importance. With DSM, we further on refer to a *Product Architecture DSM,* which we define as a component-DSM with all interactions represented as functional flows (information, energy, matter) and spatial relations. This type of architectural representation is sometimes referred to as *system architecture DSM*, *product DSM* and *component-based DSM)* (Eppinger & Browning, 2012). DSM representations are mainly used to visualize the complex lateral interactions between the product components, however, it may also be used to model hierarchical interactions, see the color-coded clusters in Figure 1.

The DSM clustering algorithm presented in Börjesson and Sellgren (2013) enables highly efficient clustering of DSM:s with arbitrary numerical values for the dependencies. DSM clustering addresses technical complexity but not strategic objectives (Blackenfeldt, 2001). Stake (2000) presented several examples of manual clustering of a DSM and a MIM, in an attempt to balance technical complexity (represented by a DSM) and business strategies (represented by a MIM). Blackenfelt (2001) presented a method on how the MD:s could be condensed into the four generic groups *Carry over*, *Commonality*, *Make or by*, and *Life cycle*, and represented the relations between those four groups as a *Component-Based DSM*, but performed no further DSM-based analysis. Williamsson and Sellgren (2016) addressed the challenge to perform trade-offs between technical complexity and company specific business strategies, and proposed a methodology referred to as *Integrated Modularization Methodology* (IMM). The core of IMM is to integrate company specific module drivers with a *Product Architecture DSM*, and then cluster the strategically adapted DSM.

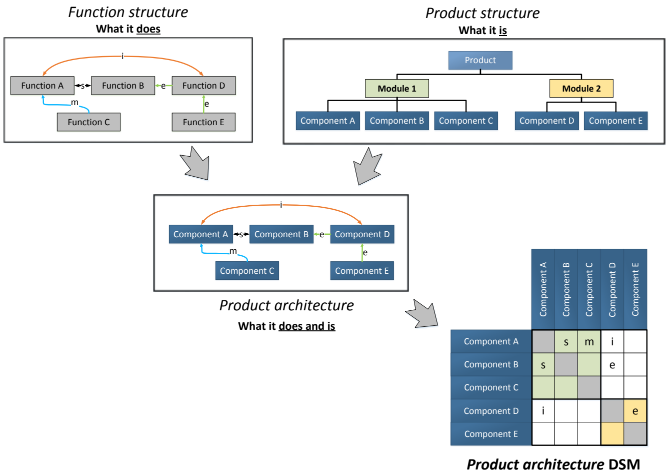


Figure 1. The type of DSM used in this paper.

A significant challenge in systems engineering is to represent and analyze the effects of architectural changes and expansions. De Weck (2007) introduced Component-Based *ΔDSM* and *Change-DSM* to represent and manage existing or future changes in complex products. A *ΔDSM* represents the difference between an original and a changed product. The *Change-DSM* contains the *change propagation paths*, i.e. how a change propagates from one component to another. A Change-DSM may therefore be used to identify components that are likely to multiply or absorb changes. No method has been proposed that can efficiently be used to analyze the difference between two DSM cluster results.

Five specific research questions are addressed in this paper:

- How can we compare multiple clustering results?
- How sensitive is DSM clustering to the relative weights of the spatial relations and the functional flows of matter, energy and signals?
- Can the DSM be used to re-engineer hidden relation weights of an architecture?
- Is IMM capable of identifying reasonable module candidates that are reasonable trade-offs between technical complexity and business strategies?
- Can IMM be used to re-engineer strategic reasons behind an architecture?

The questions are elaborated on with an industrial case. The studied case, which is presented in chapter 2, is analyzed in chapter 3 with DSM and IMM clustering, i.e. from technical complexity and module driver perspectives, and discussed in chapter 4. The main conclusions and a path for future research are given in chapter 5.

2 Case study

The presented architectural investigation was conducted at the heavy truck manufacturer Scania, which is part of Volkswagen Truck & Bus GmbH, and at KTH Royal Institute of Technology in Stockholm, Sweden. The studied gearbox was developed in-house by Scania to be a module in its modular system. Analyses of the mechanical and electrical subsystems and the embedded control software were initially performed. It should be noted that only one gearbox variant, see Figure 2, was analyzed in the presented study. This was a deliberate delimitation, since a large number of variants can be configured from the modular system.

Figure 2. An illustration of a Scania truck powertrain (left), with a heavy-duty gearbox (right)

The product architecture

Scania is frequently used as a role model for modularization. The core of Scania’s modularization principle is balanced module variants configured from a limited number of physical components and with standardized module interfaces that can be combined to satisfy different customer needs. In order to efficiently describe all potential product variants, Scania represents the modular product as a generic product structure. A generic product structure does not describe a single product variant, but rather the entire product portfolio, which internally is referred to as the *Modular Toolbox*.

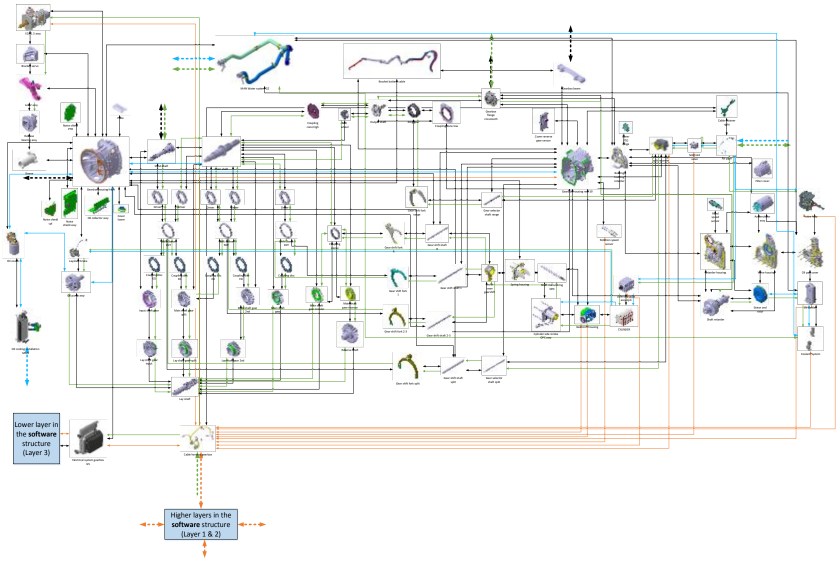


Figure 3. A component architecture diagram of the investigated gearbox architecture.

To represent the current architecture, the components and their functional purposes must first be identified. This was done by studying the physical decomposition from the generic product structure, as well as parts from the logical structure representing the electrical and software architectures. To limit the number of components, all screws, gaskets and other small parts, were not considered in the decomposition. The interactions of the targeted 94 components were represented with a *component architecture diagram* (CAD). This representation, as shown in Figure 3, visualizes the components and their functional dependencies, i.e. the principal technical function flows and spatial relations, where black indicates a *spatial relation*, green *energy flow*, blue *material transfer* and orange *information flow*.

3 Analysis method and results

The modeled product architecture was used as a test bench for studying if and how the DSM and IMM approaches may support us to find the implicit reasons (reduced technical complexity and/or business strategies) for the architecture of a highly complex engineered system, such as the targeted gearbox.

The architectural analysis method

The product architecture was represented both as a product architecture DSM and as a strategically adapted DSM to be used with the *Integrated Modularization Method* (IMM) (Williamsson and Sellgren, 2016). DSM and IMM clustering was performed with the highly efficient algorithm *IGTA++* presented in (Börjesson & Sellgren, 2013). The four types of interactions in the DSM were initially assumed to have an equal importance or weight, but the number of interaction types were added in the off-diagonal matrix cells, e.g. energy flow and a spatial relation gives an interaction value of 2. IMM clustering was performed on a strategically adapted DSM. The strategies addressed were the Module Drivers (MD:s) from the MFD modularization method.

The starting point of an IMM-based analysis is the product architecture. The relations between corporate strategies, as represented by the MD:s, and the principal solutions, i.e. the components in the DSM, are represented with the Module Indication Matrix (MIM) in the MFD method. One of the main purposes of a MIM is to identify strategically conflicting MD:s, i.e. mismatches in strategies within a module candidate. In IMM, the MIM (see upper part of Figure 4) is represented as a strategy transfer DSM (see lower mid matrix in Figure 4), with all conflicting module drivers represented with a minus sign. By operating with the strategically transfer DSM on the Product Architecture DSM, with functional interactions in the off diagonal cells, we get a strategically adapted DSM. In this transformation, all relations interfering with a minus sign gets removed from the Product Architecture DSM, while empty cells remain unchanged. In the simple example shown in Figure 4, component *D* has a conflicting module driver to the other components. According to the MFD methodology, components with conflicting module drivers should not be clustered together, in order to avoid strategic conflicts. Hence, component *D* should be separated from the other components in this case.

The module drivers, i.e. also those in conflict, for the studied gearbox were unknown. A new method was therefore needed to identify components with potentially conflicting module drivers. The core of the new method is to identify components that frequently end-up in a “wrong” cluster, compared to the existing modular architecture. This is done by comparing multiple clustering results from DSM:s with different weights for the different types of functional relations. The working hypothesis is that components which frequently end up in “wrong” clusters do that because of some (hidden) strategic aspects rather than technical. The same type of cluster comparison is also used to reveal implicitly/explicitly chosen relation weights, i.e. the technical complexity aspects behind the decisions for the existing modular architecture.

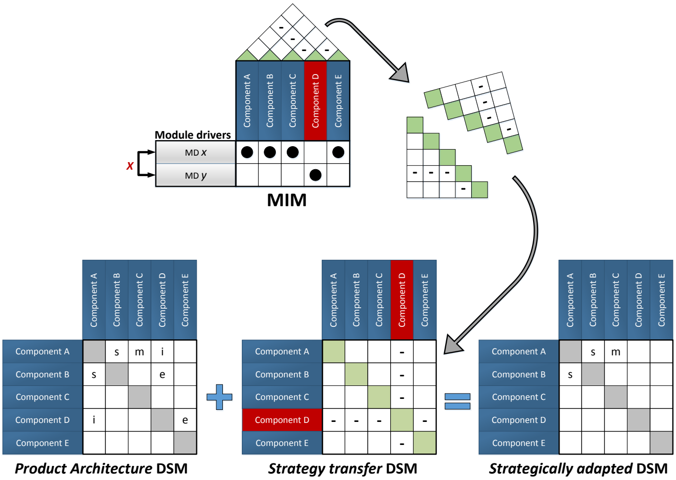


Figure 4. The integrated DSM-based product architecting model IMM.

The central representation used by the new cluster comparison method is referred to as the *Cluster Match Matrix* (CMM), which is a matrix containing a representation of a modular architecture, and the clustering results based on the different relational weight combinations. In the example seen in Figure 5, components *A*, *B* and *C* are located in one module in the original (base) modular architecture. In a similar way, components *D* and *E* are located in another separate module. Notice that the module drivers are unknown for all components in this example, i.e. we do not know that component *D* has a conflicting module driver with the other components.

The numerical values in the CMM represent the cluster number which the component is assigned to by the clustering algorithm. In the left column in Figure 5 (equal relation weights, or dependencies of the same strength), components *A*, *B* and *D* are all assigned to *cluster 1*. In a similar way, component *C* is assigned to *cluster 2* and *E* to *cluster 3*. However, since component *D* is not in the same original module as components *A* and *B*, it is marked with red, indicating that the clustered component is in the “wrong” module compared to the studied gearbox. The cluster match is finally calculated based on how many components compared to the total amount of components that are in the same module as in the actual system. With this comparison method, multiple clusters may be located in the same original module and still fulfil the criteria of a full match. For example, the original module containing component *A*, *B* and *C* is an integration of *cluster 1* and *2* in the left column in Figure 5. Hence, only components which are split from their assigned cluster, to fit the existing modular architecture, are treated as being in the wrong module. With the CMM, it is possible to compare how close a clustering result is to an existing or base modular architecture in a quantitative and repeatable way. The relation weight combination with the highest cluster match score is the one closest to the base architecture, i.e. the hidden relation weights are thus partly revealed.

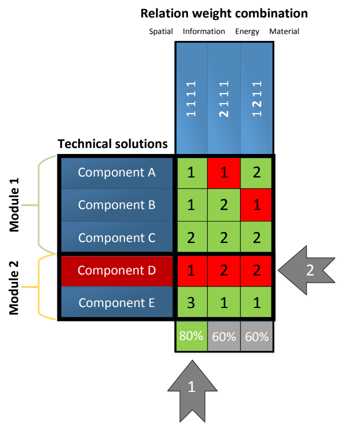


Figure 5. Example of a *Cluster Match Matrices* (CMM).

Components with conflicting module drivers may also be identified in the CMM. In the example seen in Figure 5, component *D* frequently end up in the "wrong" clusters and is therefore identified as being potentially being in conflict with the other components in the cluster, and consequently has been grouped not to reduce technical complexity, but because of some strategic reason. The same approach may also be used for situations with multiple conflicting module drivers. Furthermore, to enable CMM-base comparisons of IMM clustering with multiple conflicting module drivers, it is important to distinguish between conflicting module drivers within the same original module (based on the exiting/base architecture) and conflicting drivers external to the original module, i.e. relations should not be removed between conflicting module driver within the same original module.

Architectural analysis of the gearbox architecture

First, the studied modular gearbox architecture was represented as a component architecture diagram excluding the relations, as shown in Figure *3*. The original gearbox modules are visualized by a *Component Cluster Diagram* (CCD) in Figures 6. To identify the weight combination that generates the most similar result with the existing (expert designed) gearbox modules, an iterative approach was used. Hence, multiple clustering analyses with different combinations of relation weights were performed, followed by a CMM-based evaluation. The values used for the relation weights were 1 (functional dependency) or 2 (strong dependency). The results of these analyses are presented in Table 1. Convergence of each clustering result was found after 1500 iterations with the IGTA++ clustering algorithm in MATLAB. After performing the CMM analysis, 20 of the total 94 components were frequently (at least in 10 of 15 DSM analyses) identified to be in the "wrong" cluster. These components, marked with red in Figure 6, were identified as chosen from strategic aspects, hence, having conflicting module drivers with all other components, except components from the same original module.

As seen in Table 1, analysis #6 scored highest in the DSM based clustering. This indicates that that spatial relations and flow of information has a higher importance compared to flow of energy and material. However, this result is still far from a full match, which indicates that the existing modules were most likely not only created with an aim to reduce technical complexity.

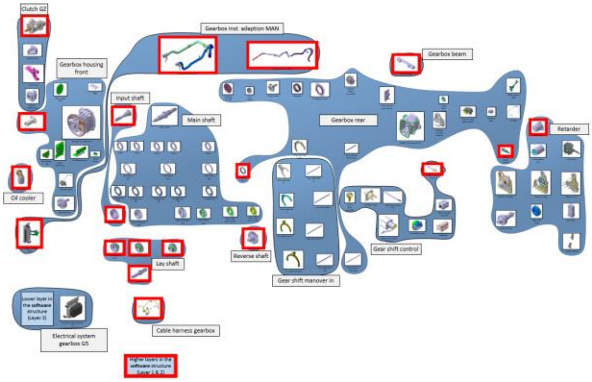


Figure 6. The studied modular gearbox architecture, including the CMM analysis result.

To include the strategic aspects, the IMM methodology was used with the relation weight combination from the base reference (best and worst CMM scores), i.e. analyses #1, #6 and #11. As shown in Table 1, all IMM results are more similar with the existing architecture since, since they got the highest score.

Table 1. Effects from different relation weights on the clustering results.

Analysis		**Relation weights**				**# Components in wrong cluster**	**Match [%]**
#	Type	Spatial	Information	Energy	Material		
1	DSM	1	1	1	1	24	74%
2	DSM	2	1	1	1	26	72%
3	DSM	1	2	1	1	22	77%
4	DSM	1	1	2	1	29	69%
5	DSM	1	1	1	2	23	76%
6	DSM	2	2	1	1	18	81%
7	DSM	2	1	2	1	27	71%
8	DSM	2	1	1	2	24	74%
9	DSM	1	2	2	1	24	74%
10	DSM	1	2	1	2	29	69%
11	DSM	1	1	2	2	31	67%
12	DSM	2	2	2	1	24	74%
13	DSM	2	1	2	2	23	76%
14	DSM	2	2	1	2	23	76%
15	DSM	1	2	2	2	21	78%
16	IMM	2	2	1	1	11	88%
17	IMM	1	1	2	2	10	89%
18	IMM	1	1	1	1	9	90%

4 Discussion

The core of the IMM method is a strategic DSM, which integrates a Product Architecture DSM with a Module Indication Matrix (MIM). The Product Architecture DSM represents technical complexity. Consequently, it is not capable of handling strategic aspects. This limitation of DSM clustering has been illustrated and confirmed in this study. Since clustering of a DSM did not result in a solution close to the investigated modular architecture at Scania, it may be postulated that the original architecture was most likely developed to provide company strategic benefits, besides from an aim to reduce technical complexity.

As seen in Table 1, the relation weights are highly important in all DSM-based analyses, since the result is largely affected by changing weight combinations, i.e. the level of dependency has a significant effect on the technical complexity. The results from the IMM analyses, on the other hand, clearly indicate that the relational weights become less important (compared to DSM clustering) when multiple strategic aspects are introduced, i.e. the solution space becomes reduced due to all constraints. In an extreme case, only the relations but not their weights will be of importance if strategies were to be considered. If more strategic aspects would be treated in the IMM analyses, e.g. if all components ending up in the wrong cluster one single time (in one analysis) would be treated as having conflicting module drivers, it would most likely be possible to reach a full match, i.e. a score of 100% in the CMM. There is also a possibility that some of the original modules were selected based on other (subjective) aspects, i.e. there may not be any technical or strategic reason behind a choice.

As earlier stated, there may be multiple conflicting module drivers, which makes it important to distinguish between conflicting module drivers within the same original module (based on the existing modular architecture) and conflicting drivers exterior to the original module, i.e. relations are not removed between conflicting module drivers within the same original module. If not considered, components with conflicting module drivers may be clustered together, even if they are not in the same original module. This will significantly lower the cluster match score.

The presented case study illustrates the importance of considering strategic aspects simultaneously with the technical complexity aspects in the architecting stage, where IMM has shown promising results. Since there is currently no accepted method on how business strategies could or should be included in DSM clustering, a new and robust methodology is clearly needed.

5 Conclusions and future work

- A *Cluster Match Matrix* (CMM) is proposed for comparing clustering results.
- Clustering a *Product Architecture DSM* is able of proposing module candidates that reduce technical complexity, but do not address strategic concerns.
- The results of the all IMM clustering analyses gained the highest cluster match scores with the existing architecture, thus IMM proposed module candidates that are most similar with the architecture as designed by domain experts.

- The presented case study indicates that the IMM methodology is capable of identifying and proposing reasonable module candidates, from both product complexity and company specific strategies points of view.
- The IMM methodology can be used for analyzing and finding the explicit and/or implicit, technical as well as strategic, reasons behind the architecture of an existing product.

The long term aim of the presented research is to develop a robust, agile and efficient modularization methodology. It is highly important to systematically investigate how the weights of the relations/dependencies in the DSM affect the clustering results, and the reasons for chosen proper weights, i.e. reliability, safety, cost and other concerns. To be able to verify, generalize, and improve the clustering results, a larger range of products and development cases have to be analyzed.

Acknowledgement

The financial support from SCANIA CV AB, as well as the open discussions with domain experts at Scania CV AB and Volkswagen Truck & Bus GmbH are gratefully acknowledged.

References

Baldwin, C.Y. and Clark, K.B. (2000). Design Rules, Volume 1, The Power of Modularity, MIT Press, Cambridge MA., USA.

Blackenfelt, M. (2001). Managing complexity by product modularisation : balancing the aspects of technology and business during the design process, Doctoral Thesis., Dept. of Machine Design, KTH Royal Institute of Technology, Stockholm, Sweden.

Börjesson, F, Sellgren, U. (2013). Fast Hybrid Genetic Algorithm for Clustering Design Structure Matrix. ASME Design Engineering Technical Conferences, Portland, OR, 2013, Paper number DETC2013-12041

Börjesson, F. (2014). Product platform design - architecting methods and tools, Doctoral Thesis., Dept. of Machine Design, KTH Royal Institute of Technology, Stockholm, Sweden.

de Weck, O. (2007) On The Role of DSM in Designing Systems and Products for Changeability. 9th International Design Structure Matrix Conference, Munich, Germany.

Eppinger, S. D. & Browning, T. R., (2012). Design Structure Matrix Methods and Applications. Cambridge, Massachusetts: The MIT Press.

Ericsson, A., Erixon, G. (1999). Controlling Design Variants: Modular Product Platforms. ASME Press, The American Society of Mechanical Engineers, New York, USA.

Erixon, G. (1998). Modular function deployment : a method for product modularization, PhD. Thesis., KTH, Dept. of Manufacturing systems, Stockholm, Sweden.

Hölttä-Otto, K. (2005). Modular Product Platform Design, Doctoral Dissertation, Helsinki University of Technology, Department of Mechanical Engineering, Machine Design, Helsinki, Finland.

Maier, J. R., Sandel, J. & Fadel, G. M., (2008). Extending The Affordance Structure Matrix – Mapping Design Structure and Requirements to Behavior. 10th International Design Structure Matrix Conference, Stockholm, Sweden.

Pimmler T.U., Eppinger S.D. (1994). "Integration Analysis of Product Decompositions", ASME Design Technical Conferences, 6th International Conference on design Theory and Methodology, DTM '94, Minneapolis, USA.
Stake, R. B. (2000). On Conceptual Development of Modular Products. Doctoral Thesis, Department of Production Engineering, Royal Institute of Technology, Stockholm, Sweden.
Ulrich, K. (1995). The role of product architecture in the manufacturing firm. Research Policy 24(3):419-440, DOI 10.1016/0048-7333(94)00775-3, ISSN 0048-7333.
Ulrich K and Tung K (1991) "Fundamentals of product modularity", Proceeding of the 1991 ASME Winter Annual Meeting Symposium on issues in Design/Manufacturing/integration, American Society of Mechanical Engineers, Miami, Florida, USA.
Williamsson, D. and Sellgren, U. (2016). An approach to integrated modularization, 26th CIRP Design Conference, Stockholm, Sweden.

Contact: D. Williamsson, Scania CV AB, R&D, Product Description Methodology, Transmissionsvägen 2, 151 48, Södertälje, Sweden, david.williamsson@scania.com

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Conceptual design of suspensions with integrated electric motors on the basis of DSM

Meng Wang, Andreas Höfer, Horst Friedrich

Institute of Vehicle Concepts, German Aerospace Center (DLR), Germany

Abstract: To successfully realize the design by combing two systems multifaceted issues need to be considered. The potential functional conflicts between the existing systems are the major unfavorable factors limiting the design. In order to better complete the design, the physical relationships of the design elements in the systems should be figured out. This paper applies the Design Structure Matrix (DSM) to design suspensions combined with electric driving unit for electric vehicles (EVs) to support the engineering functional integration process. In the process, the mutual relationships of design parameters in the systems are described by means of design structure matrices. Two engineering cases are illustrated in this paper to show this process.

Keywords: product design, electric vehicles, suspension, engineering functional integration

1 Introduction

Automobile manufacturers developing electric vehicles currently tend to convert existing conventional internal combustion engine powered vehicles into designs for electrically driven automobiles. Changing of boundary conditions and requirements associated with electric mobility are taken into consideration. Lightweight design and creating space through new package variation are new important design requirements of suspensions for electric car development.

Especially regarding automobile applications where other lightweight design methods like the usage of new materials or form optimization are already deeply exploited, further weight saving can be found by the integration of functions (Ziebart 2012).

Consequences of lightweight design by component function integration are that the resulting products are smaller, lighter and cost-efficient (Ziebart 2012) for example sandwich structures for automotive application (Kopp et al., 2009) and a metallic casting A-pillar in the front body structure (Beeh et al., 2013). A functional integration process for mechanical design refers to the realization of the functions of two systems by only one system. The design of functional integration is a very challenging task in engineering, as the designer must creatively and carefully select the design parameters (DPs) in the systems to combine and systematically evaluate the compatibility of the combined design. The physical status change of the selected DPs may affect other associated DPs, because after combination they are associated with each other; these associations may cause an unexpected performance of the design. Without design knowledge and experience, it is hard to enable the new concept with the integrated functions without appearing undesired properties.

Electric lightweight suspensions integrating drive units into the chassis (Pautzke 2010) have the advantage of reducing unsprung mass (Friedrich 2013), creating space through new packaging variations (Kriescher and Brückmann 2012) and incorporating individual wheel drives (Höfer et al., 2015).

DSM has more advantages for analyzing the interaction of existing products (Tang et al., 2008). In the automotive environment product based DSM have found various applications. DSM that helps to integrate two independent products has not yet been part of the research publications and will be outlined in the following work.

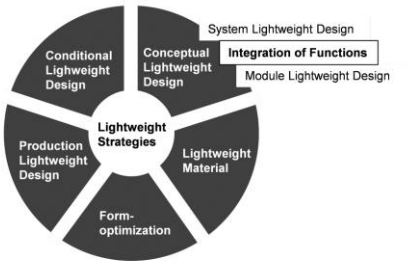


Figure 1: Overview of lightweight design strategies (Henning and Moeller 2011)

2 Fundamental of design structure matrix

A product DSM is a square matrix whose rows and columns are identically labeled with the product components, and whose off-diagonal cells indicate component interfaces (Sosa et al., 2007). The cells along the diagonal of the matrix represent the system elements. A cell can have inputs entering it and outputs leaving it (see Figure 2) representing a flow of information (see Figure 3). Off-diagonal on the lower side represents information flow that feeds the following elements; the upper side indicates that the element feeds something back upstream (Helo, 2006). The level and strength of dependency between components can be expressed by the DSM. It is able to provide critical information such as performance metrics and failure rates. This information helps project managers to identify components of importance that will require particular attention in the design process.

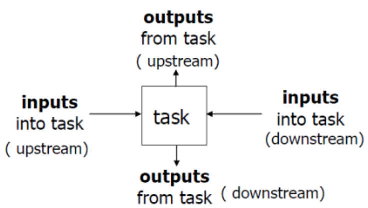


Figure 2: Inputs and Outputs of a DSM (Weck, 2012)

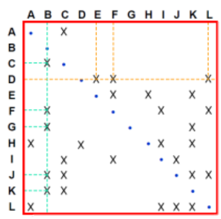


Figure 3: Flow of Information of a DSM (Weck, 2012)

Traditionally the researchers construct the DSM by interviewing the relevant technology engineers and documentations when the most important decisions about the system and the design are made (Dong und Whitney 2001).

3 Application

This paper applies DSM methods to product functional integration to realize lightweight design. The interaction among the design parameters can be studied by the DSM, which is proper tool to visualize the product architecture and relationships of the DPs. This provides the engineers another way to analyze the new concept and improves the success rate of the new concept development.

At first the independent systems should be decomposed until the design level with the common combined components. Then, the matrix flow should be investigated by the way of literature review, expert consulting, team talking or some advanced design models, with which the design hierarchy of each system is written as the matrix equation. After that, the matrices of the independent systems are arranged in one matrix equation which represents the combined system. The combined components appear in a unified matrix. Unknown flows appear on the off diagonal of the new matrix, which represent the cross effects of one system on the other system. The unknown element should be defined in order to probe the influence of the combination on the system. The DSM of the new combined concept can be derived from the DM. The DSM of the new concept improves the understanding of the intern relationships and compatibility among the design parameters. It helps to identify the important parameters, guide the engineers to pay more attention on these parameters. In the further development, engineers can take advantage of the DM and DSM to plan the engineering design.

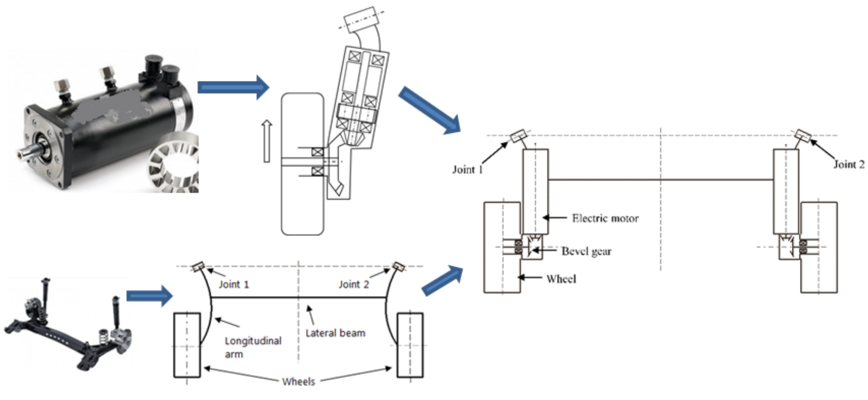


Figure 4: Plan view of the whole suspension concept

3.1 Concept design of suspension combining electric motors

The optimization of power-to-mass and torque-to-mass ratios of motors for EVs makes it possible to integrate the electric motors into suspensions. A large number of these suspensions appear in patents, papers and products, for examples the Active Wheel from Michelin (Vijayenthiran 2008), the VDO eCorner from Siemens (Sterbak 2007). The development integrating electric motors and suspension depends on designer knowledge and experience, so it is hard to enable the new concept to perform the integrated functions as expectation without appearing undesired properties. Based on the DSM, the engineering process can be carried out with awareness of the relationship of the design parameters. The concept idea under background of this project is depicted in Figure 4. The electric motor in this picture is designed to integrate to the longitudinal arm of the twist beam suspension.

DP_m:

- DP_{m1}: Rotor and stator
- DP_{m2}: Motor case
- DP_{m3}: Output shaft assembly

DP_s:

- DP_{s1}: Joints
- DP_{s2}: Arms and links
- DP_{s3}: Spring rate
- DP_{s4}: Damper coefficient
- DP_{s5}: Sprung mass
- DP_{s6}: Bushing
- DP_{s7}: Wheel alignment
- DP_{s8}: Wheel mass (interaction force of wheel and road)
- DP_{s9}: Anti-roll bar

	DP_{m1}	DP_{m2}	DP_{m3}
DP_{m1}	2		
DP_{m2}	1	2	1
DP_{m3}	1		2

	DP_{s1}	DP_{s2}	DP_{s5}	DP_{s4}	DP_{s8}	DP_{s3}	DP_{s9}	DP_{s7}	DP_{s6}
DP_{s1}	2								
DP_{s2}	1	2							
DP_{s5}	0.5	0.5	2	2		2	0.5		
DP_{s4}			2	2	2	2			
DP_{s8}	0.5	0.5		2	2	2			
DP_{s3}	1		2	2	2	2	2		
DP_{s9}	1			1		2	2		
DP_{s7}	1		1					2	2
DP_{s6}								2	2

Figure 5: decomposition and the interactions of the design elements

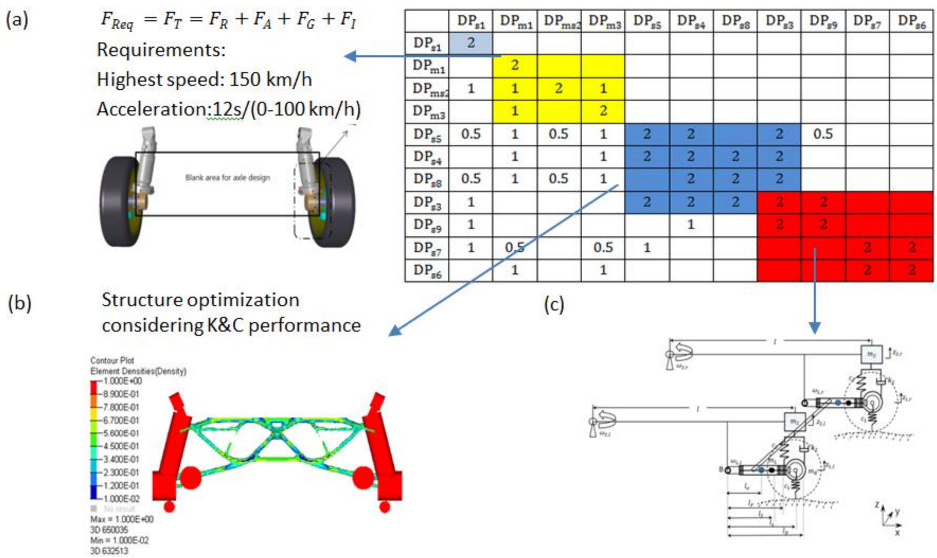
The motor and the twist beam axle are decomposed and the interactions of the design elements are expressed by the matrix in the figure 5. In each row the design parameters playing a major role are chosen as the output variables which are represented by “2”; the elements with strong interaction but not adjustable are marked by “1”; the elements with normal interaction and not adjustable are marked by “0.5”. According to the principles constructing DSM, the off-diagonal cells in the lower triangular matrix represent the forward information which affects the later element; the off-diagonal cells in the upper triangular matrix represent the feedback information i.e. the iteration.

The design elements of the two systems are arranged in one DSM, in which the design element DP_{m2} and DP_{s2} are combined to one element DP_{ms2}. The DSM is constructed and clustered according to the primary acting DPs. According to the engineering competence und facility of the research and development section the form of the result may be less different. This concept is modularized into four parts: joints, electric motors, vertical dynamics and driving stability which are distinguished by the use of four colors (see Figure 6). The interaction between the four parts is show in the DSM. From the interaction we can know that DP_{s5}, DP_{s4}, DP_{s8}, DP_{s3} and DP_{s9} must be paid more attention in the concept development, because they contained more primarily acting forward or feedback information for other DPs. From the perspective of modularity, the DP_{s3} is an important DP for both the vertical dynamics and driving stability. Therefore, it must be considered in the development of these two modules. The matrix tools allow the engineers to better understand the functional integration process and the further development process.

	DP_{s1}	DP_{m1}	DP_{ms2}	DP_{m3}	DP_{s5}	DP_{s4}	DP_{s8}	DP_{s3}	DP_{s9}	DP_{s7}	DP_{s6}
DP_{s1}	2										
DP_{m1}		2									
DP_{ms2}	1	1	2	1							
DP_{m3}		1		2							
DP_{s5}	0.5	1	0.5	1	2	2		2	0.5		
DP_{s4}		1		1	2	2	2	2			
DP_{s8}	0.5	1	0.5	1		2	2	2			
DP_{s3}	1				2	2	2	2	2		
DP_{s9}	1					1		2	2		
DP_{s7}	1	0.5		0.5	1					2	2
DP_{s6}		1		1						2	2

Figure 6: The DSM of the concept suspension

Taking the advantage of DSM, the main DPs have been determined using engineering methods. Among these parameters, the joints are the most basic element, which should be defined at first. The primary parameters of the electric motor (DP_{m1}: Rotor and stator) are calculated according to the vehicle power requirements; DP_{m3} are designed according to the transmission requirements with the condition of DP_{m1} (see Figure 7(a)). Topological structure for the concept suspension is applied under consideration of the design elements of the electric motors (see Figure 7(b)). The result of the structure optimization provides a feasible topology for the concept development, which satisfies not only the mechanical but also the K&C requirements. The design parameters in the last group are strongly related to the vertical suspension dynamics. An analytical model for rear-axle vehicle dynamics and a double lane model of road irregularities are developed (see Figure 7(c)). The parameters of the spring and damper are investigated by the analytical model of the ride dynamics.



	DP_{s1}	DP_{m1}	DP_{ms2}	DP_{m3}	DP_{s5}	DP_{s4}	DP_{s8}	DP_{s3}	DP_{s9}	DP_{s7}	DP_{s6}
DP_{s1}	2										
DP_{m1}		2									
DP_{ms2}	1	1	2	1							
DP_{m3}		1		2							
DP_{s5}	0.5	1	0.5	1	2	2		2	0.5		
DP_{s4}		1		1	2	2	2	2			
DP_{s8}	0.5	1	0.5	1		2	2	2			
DP_{s3}	1				2	2	2	2	2		
DP_{s9}	1					1		2	2		
DP_{s7}	1	0.5		0.5	1					2	2
DP_{s6}		1		1						2	2

Figure 7: Engineering process based on the design matrix

According to this process the suspension concept has been further developed into the detail design phase (see Figure 8). This suspension is mounted to the vehicle body through the bushing bearings, the springs and dampers. The lightweight linkage connects the left and right wheels and supports the lateral force on the wheels, and meanwhile it functions as an anti-roll bar with a certain torsional stiffness. The electric motors produce the drive force, which is transferred to the wheels through the gears for the whole vehicle. The reaction force on the wheels is transferred to the vehicle body through the wheel hub, the case of the gearbox, the case of the motor and the bushing bearing or the spring and damper.

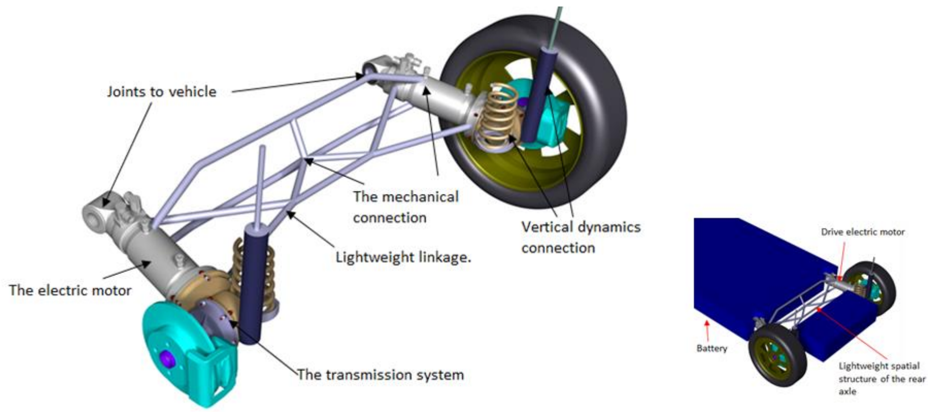


Figure 8: Plan view of the whole suspension concept

3.2 Prototype design of suspension with electric motors close to wheel hub

Another application of the DSM is in the development of the suspension with electric motor close to wheel rub. The construction of the concept is shown in figure 9. The concept has the following characteristic features (Höfer, et al. 2015) (Höfer, et al. 2016): The conventional wheel bearing in the center of the wheel is replaced by bearing elements, (1). This connects the rotating and stationary parts of the chassis. Each bearing element is fitted with six spherical roller bearings, (2), which execute rolling motion inside the rim. The space available within the wheel bearings is used to position two guide elements, (3). Vertical force absorption is implemented using two coil springs, (4), integrated into the wheel. The shock absorbers attached to the lower wheel bearing element, (6), serve as the suspension's upper impact point. A monotube shock absorber ((7), partly hidden) is used. The lateral forces induced in the wheel contact patch are passed on to the two wheel bearings via two lateral guide rails, (8). The bearing seal (9) is a labyrinth seal produced by additive manufacturing.

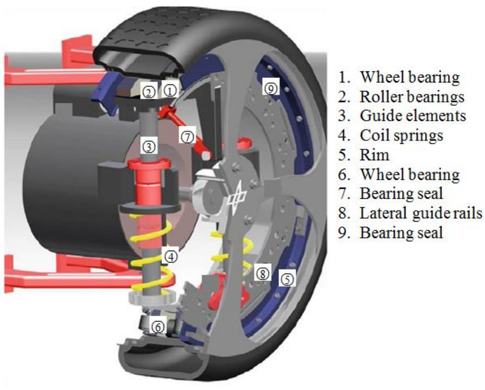


Figure 9: Lightweight suspension construction

The DSM of this suspension concept is show in figure 10, which is the first level DSM structure. It can be analyzed that the wheel bearings and the guide elements have the most interactions with other design elements. The rim, lateral guide rails and bearing seal are influenced by electric motors, guide elements and bearings.

	DP_{m1}	DP_{s1}	DP_{s6}	DP_{s2}	DP_{s3}	DP_{s5}	DP_{s8}	DP_{s9}	DP_{s4}	DP_{s7}
DP_{m1}	2	1								
DP_{s1}		2		1			0.5			
DP_{s6}			2	1			0.5			
DP_{s2}		1	1	2	2					
DP_{s3}	0.5	1	1	2	2	2	2			
DP_{s5}		1	0.5		2	2	2	2		
DP_{s8}	1	0.5	0.5	0.5	0.5	2	2	2		
DP_{s9}	1				1	2	2	2		
DP_{s4}	0.5	1		1	0.5				2	2
DP_{s7}	0.5	1		1	0.5				2	2

Figure 10: General DSM of the lightweight suspension

The design elements and the DSM are further developed. The development process of the physical concept can be seen in the figure 11. The rim and lateral guide rails are designed and validated by using FEM; the electric motor, springs and dampers are defined in the multibody dynamics and they are further optimized on the basis of this multibody simulation. According to the results of the design elements, a prototype suspension is built. The design parameters and the relationship among them will be further validated and developed.

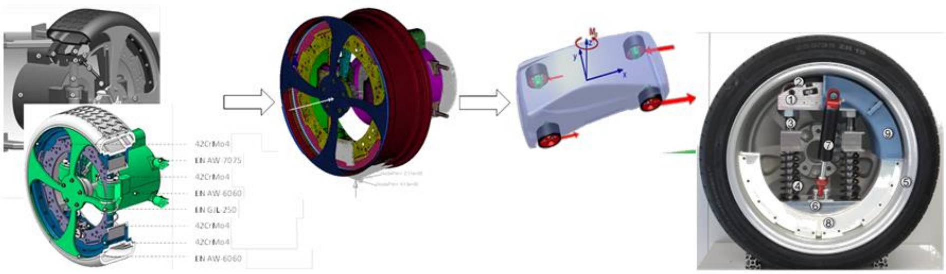


Figure 11: Design (CAD) and simulation (FEM and MBS) of the lightweight suspension

4. Conclusion

This paper has devoted to apply the DSM in the concept development which aims to integrate the function of two systems on the purpose of structure lightweight. The design process is expressed by matrix equations on the basis of DSM. The DSM helps the designer to understand the relationship between the system functions and parameters and the interaction among the design parameters in the design process. Two case applications are illustrated in this paper. The matrix among DPs benefits the designer to evaluate and compare the concepts, while the DSM helps the engineer identify the important DPs and assist the engineering process. Based on the DSM, the relationship between the systems to be combined and the relationship among the DPs are shown in the design matrix. The visible relationship enables the functional integration design be managed in a more effective and logical manner than the traditional concept–test design way.

The principle of this approach can serve as a theoretical foundation for the future design research. For example, a database based on the design matrix for the automotive components can be built. The possibility combining the components to achieve lightweight design can be studied by this approach, which is a time-saving and cost-saving process.

Reference

Beeh, E., Kriescher, M., Deißer, O., Kopp, G., and Friedrich, E.H., 2013. Crash Sicherheits Potentiale durch leichte, funktionsintegrierte Fahrzeugstrukturen. 4. ATZ-Fachtagung: Werkstoffe im Automobilbau. Stuttgart.

Weck, D.O., 2012. Design Structure Matrix. Massachusetts: MIT Lecture 04.

Friedrich, H.E., 2013. Leichtbau in der Fahrzeugtechnik. Stuttgart: Springer Vieweg.

Helo, P.T., 2006. Product Configuration Analysis with Design Structure Matrix. Industrial Management & Data Systems 106, 7, 997-1011.

Henning, F., and Moeller, E., (2011). Handbuch Leichtbau: Methoden, Werkstoffe, Fertigung. Carl Hanser Verlag München Wien.

Höfer, A., Zeitvogel, D., Friedrich, H.F., and Wiedemann, J., 2015. Ganzheitliche Betrachtung von Fahrwerk, Antrieb und Fahrdynamikregelung. Automobiltechnische Zeitschrift (ATZ) 04, 68-73.

Höfer, A., Schlamp, D., Wang, M., Kopp G., Kopp G., and Friedrich H., 2016. Development and Performance of an In-Wheel Suspension Concept with an Integrated Motor. FISITA 2016 World Automotive Congress. Busan, Korea.

Kopp, G., Friedrich, H.E., Kuppinger, J., und Henning, F., 2009. Innovative Sandwichstrukturen für den funktionsintegrierten Leichtbau. ATZ Automobiltechnische Zeitschrift 4, 111.

Kriescher, M., and Brückmann, S., 2012. Funktionsintegrierter Extremleichtbau in Sandwichbauweise. ATZ. Hamburger Karosserietage. Hamburg.

Friedbert, P., 2010. Radnabenantriebe: Studie zur Abschätzung der Verfügbarkeit, des Entwicklungsstandes und des Potentials von Radnabenantrieben in der Elektromobilität. Aachen: Shaker Verlag.

Dong, Q., and Whitney, D., 2001. Designing a requirement-driven product development process. Proceedings of the 13th international conference on design theory and methodology (DTM2001). Pittsburgh, Pennsylvania, USA,.

Sosa, E.M., Browning, T., and Mihm, J., 2007. Studying the Dynamics of the Architecture of Software Products. Proceedings of the ASME 2007 International Design Engineering Technical Conferences. Las Vegas: ASME.

Sterbak, R., 2007. Technology for the Environment – Environmentally Friendly Transportation Lowering Emissions. Siemens.. http://www.siemens.com/innovation/en/publikationen/publications_pof/pof_spring_2007/technology_for_the_environment/green_transportation.htm.

Tang, D.B., Zhang G.J., and Dai, S., 2008. Design as integration of axiomatic design and design structure matrix. Robotics and Computer-Integrated Manufacturing 25, 610-619.

Vijayenthiran, V., 2008. Michelin's Active Wheel technology in detail. MotorAuthority. http://www.motorauthority.com/news/1030025_michelins-active-wheel-technology-in-detail.

Ziebart, J.R., 2012. Ein konstruktionsmethodischer Ansatz zur Funktionsintegration. Technische Universität Braunschweig.

Part IV: Using Data

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Data-Based Development of an Agent-Based Simulation to Support the Design of Bicycle-Sharing Systems

Christoph Hollauer[1], Christopher Lang[1], Julian Wilberg[1], Jörg Weking[2], Christian Dengler[3], Markus Böhm[2], Helmut Krcmar[2], Boris Lohmann[3], Mayada Omer[1]

[1]Laboratory for Product Development and Lightweight Design, Technical University of Munich (TUM)
[2]Chair for Information Systems, TUM
[3]Chair of Automatic Control, TUM

Abstract: Bicycle-Sharing Systems are emerging as alternative modes of transportation, successfully combining product and service aspects similar to those of other Product-Service Systems. Since such systems are influenced by a number of factors during their operation, identifying ways to manage the dynamic complexity during the operational phase is desirable. In this paper, we present an approach using agent-based modeling in combination with data analytics of system usage data to analyze the impact system architecture changes would have on overall system behavior.

Keywords: Product-service systems, bicycle sharing, agent-based modeling

1 Introduction

With the expected increase in the use of autonomous vehicles in the near future, major shifts in urban mobility and infrastructure are already challenging traditional forms of transportation, with sharing models in particular gaining a larger influence. Similar to heavily marketed car-sharing services, bicycle-Sharing Systems (BSSs) have rapidly increased in recent years, typically providing short-term rental of bikes (Büttner et al 2011, p. 10). These systems offer numerous benefits, including reduced emissions and congestion, improved overall health, and extended public transportation for what is known as the "last mile" (Shaheen et al 2010).

Such integrated Product-Service Systems (PSS) require complex design processes, thereby increasing the number of involved disciplines (Schenkl et al 2013). More specifically, multiple design aspects must be considered, including bike design, access management, legal regulations, revenue streams, and ongoing maintenance. Each design decision in setting up a BSS has a significant impact on the user experience, associated costs, and sustainability. Further, many of these choices affect one another, thus increasing the complexity and creating a highly intricate system. In brief, decision support is required for stakeholders developing and operating such complex systems (Rouse 2007).

Given the above, our objective in this paper is twofold. First, we present a formalized methodology for managing the complexity of designing a bicycle-sharing operating model with the help of approaches for managing structural and dynamic complexity. Second, based on the structural elements of the BSS architecture, we present a comprehensive agent-based model (ABM) to analyze and improve aspects of our model. We developed the methodology described in this paper primarily for a hybrid (combining station-based and free-floating) BSS in Germany; however, our model is applicable to other systems.

2 Research context and methodology

The research results that we present in this paper stem from a joint research project with a Munich-based BSS operator. The focus of this project was to develop a new BSS targeted at physically impaired users. A core aspect of our research addresses the operating model of said system, including an analysis of the existing BSS operating model.

We rely on the design research methodology defined by Blessing and Chakrabarti (2009) as a foundation for our present research. The *Research Clarification* was conducted in cooperation with an industry partner and based on an initial literature review. Within the *Descriptive Study I*, we investigated the current system and planning practices of the industry partner. Further, we performed a literature review on existing approaches for modeling, analyzing, and designing BSSs. Within the *Prescriptive Study*, our methodology was iteratively developed, applied, and improved upon within the research project context. We collected regular feedback from our industry partner (support evaluation) and a concluding evaluation workshop was held during the *Descriptive Study II*.

3 Background and related work

The approach that we propose in Section 4 below is based on a variety of concepts in complexity management and simulation modeling, which we summarize in the subsections that follow.

3.1 Modeling structural and dynamic complexity

To model system architectures, we can turn to either Domain-Specific Languages or universal languages (e.g., SysML) (Kerzhner & Paredis 2009). Both, graphs or matrices can be used to represent structures of complex systems; both representations are equivalent and transferrable to one another (via adjacency matrices) (Tittmann 2003). Matrix-based representations are, for example, Multiple-Domain-Matrices (MDM) that consist of Design Structure Matrices (DSMs) and Domain Mapping Matrices (DMMs), which represent intra-domain and inter-domain dependencies, respectively (Lindemann et al 2009).

Simulation models can be used to model dynamic complexity, in particular by recreating the behavior of real-world processes or systems over time (Banks et al 2005, p. 3). To model complex real-world systems, Borshchev (2013, p. 37) describes three modeling paradigms, primarily differentiated by their degree of abstraction; these are Discrete Event Modeling, System Dynamics, and ABM.

We selected ABM as the foundation for capturing the temporal behavior of the investigated BSS. Our approach here is based on the bottom-up modeling of the behavior and interactions of individual agents, such as individuals or vehicles. Therefore, it allows observing emergent behavior (Macal & North 2010, p. 1), which often arises in complex systems composed of autonomous subsystems with different objectives (Rouse 2007). This is realized by modeling individual behavior rules (e.g., using State Charts) (Bonabeau 2002, p. 1).

3.2 Existing approaches and research gaps

To identify existing approaches for modeling and analyzing BSSs, we investigated the prevalence of the following aspects in the literature:

- **BSS operating model and description** are defined as the architecture of the BSS during the operational phase, containing all consciously induced and controlled organizational and physical boundary conditions to achieve the system objectives (cf. e.g., Lathia et al 2012, Büttner et al 2011).
- **Data analysis and demand forecasting** are described in detail in Fishman (2016), which provides a review of current approaches regarding BSS and data analysis.
- **Redistribution and user incentives of free-floating systems** are minimally covered in the literature, with only one identified publication that investigates a free-floating system, cf. Reiss & Bogenberger (2016).
- **Station and system planning** primarily addresses the expected long-term demand and corresponding placement and sizing of stations. No literature has been identified that addresses the planning of free-floating systems, e.g., the definition of the business area.
- **The use of pedal electric cycling (pedelecs) in a BSS** is a trend in BSS design, but no publications could be identified that describe the use or long-term system behavior in comparison to regular bikes.
- **Costs and cost structures in BSSs**, often described as optimization targets (e.g., Hu & Liu 2014), were rarely addressed explicitly.
- **Additional approaches for modeling and simulation of a BSS** address only singular aspects but cover the modeling paradigms presented in Section 3.1 above.

While an extensive reproduction of the current state-of-the-art is not the focus of our present paper, based on our literature review, we have identified a lack of research regarding the analysis of hybrid and free-floating BSSs using data- and simulation-based methods. Most literature focuses on station-based systems in which bikes can only be rented and returned at fixed stations. Further, there is no known approach for capturing the overall system architecture with a suitable simulation. Therefore, our work addresses the following four objectives:

- Provide an extensive analysis of a hybrid BSS, including pedelecs
- Define an extensive BSS architecture for the corresponding operating phase
- Show a systematic alignment of problems that may arise during the operations phase with components of the system architecture
- Develop an ABM for evaluating BSS architecture changes

4 Methodology for data-driven development of the ABM

To enable the creation and extensive analysis of a BSS using ABM, we developed a corresponding procedure. In this section, we first provide an overview of this methodology, then we illustrate the individual steps using the particular BSS from our research project. The procedure developed within our work, depicted in Figure 1, is adapted from the methodology described by Hollauer et al (2015). Our approach further adapts the Knowledge Discovery in Databases process formulated by Fayyad et al (1996) to integrate usage data of the BSS into the model-building process. Within our methodology, we build structural models of the system architecture and investigate the model using matrix-based approaches that are later used as a basis for developing the dynamic ABM. Therefore, methods of complexity management (cf. Section 3.1) are integrated within the problem-field analysis step (3) shown in the figure.

As indicated in Fig. 1, our methodology first analyzes existing data related to the usage of the investigated BSS, the results of such analyses being used to describe system behavior and identify the impact of various influencing factors, e.g., weather (1). In parallel, the BSS architecture elements that are relevant to its operational phase are identified and described (2). To support this activity, we developed a general framework for the operational BSS architecture. Next, a problem-field analysis is conducted in which existing everyday operational problems are matched to corresponding system architecture elements in a DMM (3). Based on the knowledge acquired from both the problem-field analysis and the usage data analysis, recommendations for action are then discursively derived; these actions are focused on improving the everyday operations of the BSS (4). Finally, the information acquired from the usage data analysis is used to develop an agent-based simulation of the BSS, which then allows for testing how the recommendations affect overall system performance (5).

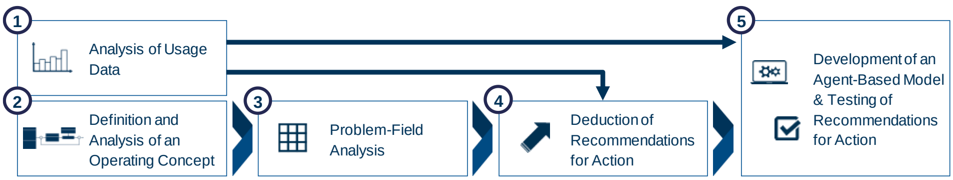


Fig. 1: Basis and developed approach for constructing an ABM of a BSS

4.1 Analysis of usage data for ABM development (step 1)

Within the modeling process, data is initially used to gain a better understanding of the behavior of the real system, then later used during the creation of the model to define individual parameters within the model based on concrete data. Fig. 2 illustrates the procedure that we followed to integrate the data within the modeling process.

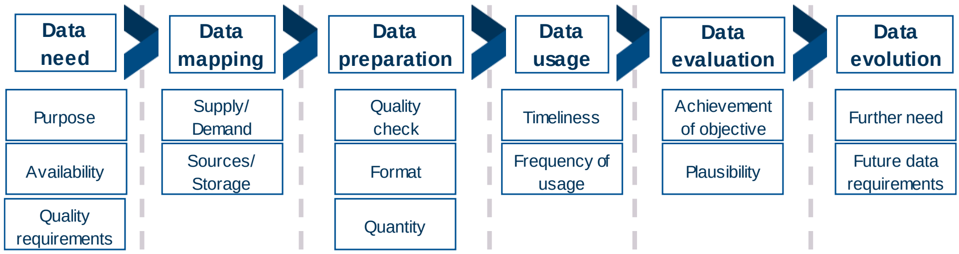


Fig. 2: Our seven-stage procedure for data integration during the modeling process

First, the *data need* for creating the model is determined, after which the sources and storage of the data is *mapped*. Subsequently, the data is *prepared* for analysis and eventually *used* within the modeling process (i.e., analyzed and insights derived, e.g., the median rental rate of a station per each hour of the day). Next, the data is *evaluated* for suitability and possible *needs for further data* are identified.

For our analysis, we used the *Tableau* software to analyze a database containing 621,579 data points describing individual bike rentals ranging from October 2015 to June 2017. Each data point contained, inter alia, data regarding the time and location of the start and end of each trip, the respective bike used, specific customer ID, the price tier of the customer, and the calculated price of the trip. We focused our analysis on the following aspects:

- Rental frequency per bike and per day
- Distribution of rentals per day, with working and weekend days noted
- Distribution of rentals per station within a single day
- Distribution between free-floating and station-based rentals (i.e., pick-up and return mode)
- Development of the number of stations and overall parking spaces over time
- Geographical heat map of rental distribution per postal code area
- Geographical heat map of rental distribution compared with selected price tier
- Ratio of pick-ups and returns per postal code area
- Distribution of trip duration per day
- Distribution of trip distance per day
- Average trip duration per day and per month
- Share of round trips per year and per month
- Average trip duration and distance per price tier
- Number of booked packages per month
- Number of customer service requests per month
- Number of repairs per month
- Number of offline hours of stations per month
- Number of active customers and number of rentals
- Growth of customer base per month
- Correlation between weather influences (e.g., mean temperature, average amount of sunshine, average of cloud cover, average rainfall, mean wind speed, average snowfall, etc.) and number of rentals aggregated per month and per day

As an example, we determined that trips beginning and ending with a free-floating bike position were the dominant trip type, with 47–66% over the investigated period of time. Data regarding the overall and hourly rental distributions and ratios between pick-ups and returns were subsequently used to define the rental parameters for the individual station agents of the ABM. Further, the influence of weather parameters on system usage were calculated and augmented with historical weather data. Note that the analysis we conducted allowed for only tentative insights into correlations between weather conditions and system behavior. Correlation values regarding weather influences on system behavior contained high variances, in particular during the winter months.

4.2 Definition and analysis of architectural elements of the operational phase (step 2)

To manage the architectural complexity of the BSS and derive the inherent dependencies, we created a general framework for structuring the BSS architecture relevant to its operational phase; this construction was based on our previous work and is illustrated in Fig. 3. The resulting model includes all factors that the operator can directly influence, as well as external influences, unforeseeable effects, and so on. The depicted structure follows the logic of a control feedback loop (Lunze 2010). On the left, the *System Input* defines the desired system states and intended usage, which can be divided into *Strategic Management* and *Infrastructure*. In day-to-day use, the system can shift to undesired states, e.g., a malfunction or theft of one or more bikes. Therefore, *Corrective Measures* are required to lead to the desired system state, which contains the item *Service*. The last block, *Usage*, contains the elements *Perturbation* and *System Utilization*. While the latter describes normal and intended usage by customers, *Perturbation* contains all possibly unexpected influences on the system outside of the control of the operator. Note that the structure does not include exogenous factors, such as legislation, the availability of external transportation modes, or traffic route infrastructure.

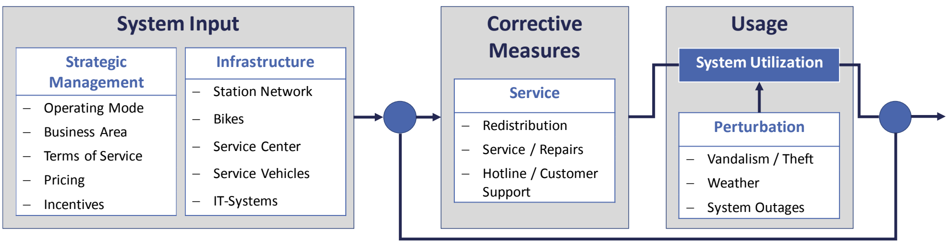


Fig. 3: Generic framework to support modeling the system architecture during the operational phase and identified system architecture elements

4.3 Problem-field analysis (step 3) and deduction of recommended actions (step 4)

The problem-field analysis depicted in Fig. 4 maps identified problems in the BSS operational phase onto BSS architecture elements via a DMM. Here, the DMM can then be converted into a DSM via matrix multiplication to identify connected problem clusters via common architecture elements.

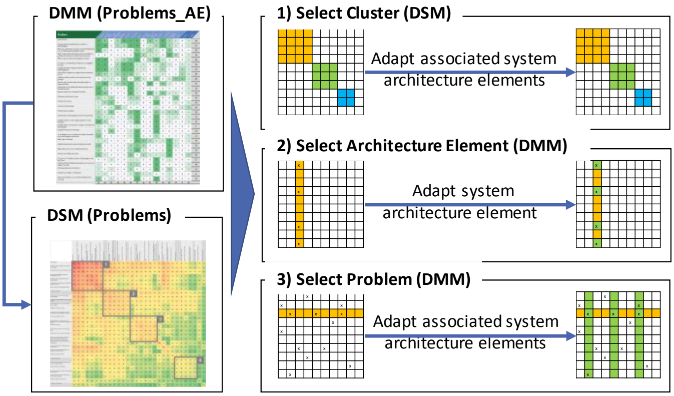


Fig. 4: Structure-based problem-field analysis and derivation of recommended actions (note that we did not intend to have the matrices on the left be readable)

Recommended actions can then be deduced via the three following strategies: (1) selecting a problem cluster to be addressed; (2) selecting an individual architecture element and subsequently analyzing the impact architecture changes have on associated problems; or (3) selecting a problem, then subsequently varying the associated architecture elements and analyzing the impact the change has on all associated problems. Within our specific application, we manually identified four key problem clusters: (1) station fulfillment and redistribution, (2) returning of bikes, (3) overall state of bike maintenance, and (4) the user app used for bike rentals.

4.4 Development of the ABM and testing of architectural changes (5)

The system architecture (step 2) and analysis results (step 1) are used to define the architectural elements required to realistically represent the BSS in an ABM, which we implemented using the *AnyLogic* software. As the BSS architecture indicates, the number of elements is quite high, thus not all elements can be implemented simultaneously. In Fig. 5, we illustrate how various elements of the BSS architecture were implemented in our simulation prototype. While some elements could be implemented in the form of agents, others had to be implemented via system or behavioral parameters. For example, the *Operating Mode* element describes how bikes can be rented and returned, but this element cannot be implemented via an agent; instead, it must be implemented as a behavioral parameter of another agent, i.e., the user.

BSS element	Implementation in the Simulation	Agents
Strategic Management		
Operating Mode	Station Based, Free-Float and Hybrid	
Business Area	Business Area of BSS	
Terms of Service	Partial Replication of the Terms of Service of BSS	
Pricing	Not Implemented	
Incentives	Not Implemented	
Infrastructure		
Station Network	Real Station Network of BSS	*Station*
Bikes	Current Number of Bikes at BSS as well as planned Expansion Number	Bike
Service Center	Implemented at the current location of the service center of BSS	Servicecenter
Service Vehicle	Current Number and Capacity of BSS Service Vehicles	Servicetruck
IT-Systems	Not Implemented	
Service		
Redistribution	Implemented (incl. variable	

Fig. 5: Implementation of BSS architecture elements in the ABM (excerpt)

Fig. 6 illustrates the basic functional logic of our simulation and the interactions of the agents. The main agent represents the environment within which all other agents act. It contains a map as well as the boundaries of the BSS operating area divided by postal code areas. Interacting agents are the users, bikes, stations, service center, and service trucks. To measure system performance, we introduced Key Performance Indicators into the main agent. As one example, each successful bike rental results in an increase of user satisfaction points, whereas each unsuccessful trip results in negative points. We used the ratio between these positive and negative points as a measure of user satisfaction.

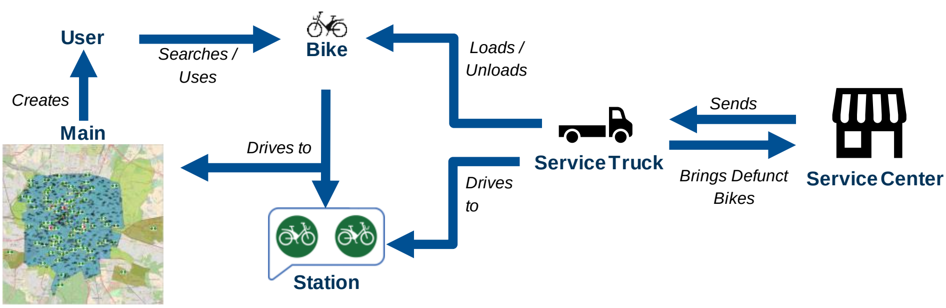


Fig. 6: Functional logic of the ABM, showing the relationships between agent classes

We used the ABM to run tests with a variety of input parameters. The ABM was used to test four specific scenarios for the following recommended actions and their impact on the respective performance metrics (step 3):

1. A reduction of the system to a purely station-based system with nine additional stations derived from real future extension plans
2. A reduction of the free-floating operating area to 10 central postal code areas with a purely station-based system outside this area

3. An increase of 100% in the number of bikes
4. An increase in bike robustness

To test the derived architectural changes, we simulated a one-week period from April 1 to April 8. Longer simulation runs were not possible due to performance constraints. Scenario 1 from above resulted in a dramatic loss of user satisfaction due to the incomplete station network that cannot compensate for the loss of free-floating bikes. Conversely, scenario 2 resulted in very high user satisfaction since users looked for bikes close to well-known stations and the central area was well-saturated with bikes. Conversely, the increased usage and high degree of full stations overloaded the repair cycle, thereby resulting in an increase in damaged bikes since repair trucks were only allowed to re-integrate bikes into the system via stations. Scenario 3 similarly resulted in high initial user satisfaction (97%) followed by a subsequent drop to 88.5% due to the increase in the number of damaged bikes. This increase occurred because the damage model within the simulation calculates the probability of a defect in relation to the number of rentals per bike. The high number of defects eventually overloads the repair cycle, which has not been adapted to the increase in bikes. Finally, Scenario 4 produced increased user satisfaction while simultaneously reducing the repair effort within the simulated period.

5 Interview-based evaluation and discussion

To evaluate the applicability and usefulness of the methodology and ABM, 10 employees of the BSS operator (i.e., the department for strategic planning) and contracted companies were presented with a demo of the approach and asked to fill out a questionnaire based on a five-step Likert scale. Results of this evaluation were generally positive, indicating that our methodology allowed for a structured approach for capturing the current state of the system and systematically searching for measures that can both improve system performance and support planning of future system expansion. From Section 4 above, the application indicated that combining the framework, analysis of usage data, problem-field analysis, and ABM can together help increase the understanding of the current system architecture and systematically deduce potential avenues for improvement. Success and influencing factors on different levels (e.g., customer satisfaction, service performance, profits) can thereby be subject to targeted analyses. In particular, the possibility of analyzing and comparing different architecture configurations should be viewed as a strength that stresses the principal usefulness of the ABM for the design of BSSs and PSSs in general.

Nonetheless, the complexity and effort involved in creating and maintaining the ABM are considerable. Our presented ABM is incomplete in regards to the modeling of external influencing factors, such as alternative modes of transportation, competition, integration within the BSS, as well as such influences as legislature or long-term climate changes. In addition, the interviewed employees noted that our methodology focuses more on the improvement of existing systems and not necessarily the design of new system architectures. The level of detail of the BSS architecture elements varied substantially, and the traceability of the overall process could have been higher.´Therefore, the ABM could only be used to validate limited architectural changes since the results strongly depend on the underlying logic and assumptions of the model (e.g., in regards to the damage model). Further, only a limited time period could be simulated; therefore, long-term effects could not be observed in our simulation as of yet.

6 Summary and future work

In this paper, we presented an approach for modeling and analyzing the dynamic complexity of a PSS applied on a real-life BSS. Our approach utilizes usage data and methods for modeling and analyzing the structural complexity. We applied our approach within a development project focused on the advancement of the current BSS. Our approach was positively evaluated via concluding expert interviews, highlighting the potential for increased system understanding. Conversely, the complexity involved in creating the ABM was considerable, and the potential to validate design decisions is still rather limited. One key area of improvement is extending support to improve the handling of this complexity during the modeling process. One way to address this is to automatically transform structural models into ABM simulations via code generation. Configurable ABMs could further reduce the modeling efforts required since they could easily be adapted, e.g., to different geographical boundary conditions. Further, the ABM could be extended to cover a simulation of a business model by investigating cost and revenue mechanisms more closely, thereby optimizing profits. As design support has only been applied to a single case study, further evaluation of a number of case studies is required to refine our proposed design support.

Acknowledgement

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 processes – Integrated development of product-service systems based on technical products’.

References

Banks, J., Carson II, J. S., Nelson, B. L., & Nicol, D. M., 2005. Discrete-Event System Simulation: Fourth Edition. Pearson.

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

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences, 99(suppl 3), (S. 7280-7287).

Borshchev, A. (2013). The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6. AnyLogic North America

Büttner, J., Mlasowsky, H., Birkholz, T., & ... (2011). Optimising Bike Sharing in European Cities-A Handbook.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI magazine 17(3), S. 37-54.

Fishman, E. (2016). Bikeshare: a review of recent literature. Transport Reviews, S. 92-113.
Hollauer, C. W. (2015). A Matrix-based Framework to Support Dynamic Modeling of Sociotechnical Systems. Proceedings of the 17th International DSM Conference Fort Worth (Texas, USA), 4-6 November 2015 (p. 19). Carl Hanser Verlag GmbH Co KG.
Hu, S.-R., Liu, C.-T.(2014). An optimal location model for a bicycle sharing program with truck dispatching consideration. Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference , S. 1775-1780.
Lathia, N. A. (2012). Measuring the impact of opening the London shared bicycle scheme to casual users. Transportation research part C: emerging technologies, 22, S. 88-102.
Lindemann, U.; Maurer, M.; Braun, T., 2009.Structural Complexity Management - An Approach for the Field of Product Design. Berlin: Springer 2009. ISBN: 978-3540878896.
Lunze, J. (2010). Regelungstechnik 1. Springer Verlag.
Macal, C. M., 2010. Tutorial on Agent-Based Modelling and Simulation. Journal of Simulation, 4(3), S. 151-162.
Reiss, S., Bogenberger, K., 2016. Optimal bike fleet management by smart relocation methods: Combining an operator-based with an user-based relocation strategy. Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference, S. 2613-2618.
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
Schenkl, S. A.; Behncke, F. G.; Hepperle, C.; Langer, S. & Lindemann, U., 2013. Managing Cycles of Innovation Processes of Product-Service Systems. In: Lai, L. (Hrsg.), Tagungsband SMC 2013. IEEE International Conference on Systems, Man, and Cybernetics, Manchester (Großbritannien), S. 918–923.
Shaheen, S., Guzman, S., & Zhang, H., 2010. Bikesharing in Europe, the Americas, and Asia: past, present and future. Transportation Research Record: Journal of the Transportation Research Board, S. 159-167
Tittmann, P., 2003. Graphentheorie. Eine anwendungsorientierte Einführung mit zahlreichen Beispielen und 80 Aufgaben. Leipzig: Carl-Hanser. ISBN: 978-3446223431.

Contact: C. Hollauer, Technical University of Munich, Laboratory for Product Development and Lightweight Design, Boltzmannstraße 15, 85748 Garching bei München, Germany, +49 89/28915136, +49 89/28915144, christoph.hollauer@tum.de

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Understanding task execution time in relation to the multilayer project structure: Empirical evidence

Sebastiano A. Piccolo[1], Jakob Trauer[2], Julian Wilberg[2], Anja M. Maier[1]

[1]Technical University of Denmark, Department of Management Engineering
[2]Technical University of Munich, Department of Mechanical Engineering

Abstract: Estimating task execution time is essential for planning and managing engineering projects. Many process scheduling and optimisation tools and methods require precise task execution time estimates. However, estimates are often too optimistic, potentially harming the usefulness of such tools. In this paper, we develop a methodology to aggregate multiple data sources into a Multiple Domain Matrix and show that its structural properties correlate with task execution time. Specifically, using data from a real-world engineering case, we show that the *size* of a task, the *number of people* assigned to it, and the number of *interfaces* directly correlate with task execution time. We discuss how these measures are available during the planning stage of the process and how people can use them to obtain better estimates.

Keywords: multilayer networks, MDM, task execution time estimation, design project, data science

1 Introduction

In late 2005, the Hamburg Parliament decided to start the construction of a new concert hall in the centre of the city – the "Elbphilharmonie". Several independent consulting companies estimated € 186.7 million in line with a feasibility study for the completion of this ambitious construction project. The targeted opening date was the 30th March 2010 (Parliament of the Free and Hanseatic City of Hamburg, 2014). By the 4th of November 2016, the building was officially finished – a delay of more than six years with a budget overrun of more than € 679 million. The "Elbphilharmonie" is just one example of project mismanagement and exemplifies the potentially catastrophic consequences of unrealistic and undersized estimations of budget and time. Good time estimates are crucial to project success (Murmann, 1994; Thamhain and Wilemon, 1986) and many tools have been developed in the attempt to support experts in their estimates and project planning (Bashir and Thomson, 2001; O'Donovan et al., 2005).

Why do experts underestimate project completion time? Humans have a tendency to underestimate the difficulties of the tasks for which they are providing estimations (Flyvbjerg, 2006; Kahneman and Lovallo, 1993). In addition, the tasks to estimate are often considered in isolation without a systemic understanding of the whole (Kahneman and Lovallo, 1993). For this reason, calls to action for using historical data to correct and/or inform time estimations have been made (Flyvbjerg, 2006; Halkjelsvik and Jørgensen, 2018).

In this paper, leveraging the intersection between engineering design and network science, we combine three different data sources from a large-scale design project of a biomass power plant in order to understand task completion time in relation to the project structure. We show that task completion time correlates positively with the number of documents produced within the scope of a task, the number of tasks to which a task is connected, and the number of people assigned to it. Our results are in line with previous research and show that the analysis of historical or archival data can generate a useful understanding of factors that can affect a project. We discuss how such an approach can offer a more global view and support project planners in estimating task completion time.

After a brief overview of the background and related literature (section 2), we introduce the datasets and the analysis methods (section 3). We report the results (section 4) and discuss their implications, connections with extant literature, and avenues for future research (sections 5 and 6).

2 Background

Estimating project completion time is a crucial task in the life of a project. Time estimates are important not only for financial reasons such as to present the project to possible investors, time estimates are an input variable of many project management tools. Project scheduling techniques such as the Process Evaluation and Review Technique (PERT) and the Critical-Path Method (CPM) (Project Management Institute, 2017) or techniques based on Design Structure Matrices (DSM) (Eppinger and Browning, 2012) require entering completion time for each task. As a result, errors in the estimations of tasks completion time can seriously harm the subsequent project planning and management.

Despite the models developed (for instance, Bashir and Thomson, 2001; Srinivasan and Fisher, 1995), expert estimation seems to be the most common way to estimate effort and completion time (Halkjelsvik and Jørgensen, 2018; Project Management Institute, 2017). On the one hand, expert estimation has its advantages, as experts may have important domain knowledge that the model does not include (Jørgensen, 2004). On the other hand, expert estimations are inherently prone to human and situational biases (Jørgensen, 2004; Kahneman and Lovallo, 1993) that make them too optimistic. This optimism bias happens as experts tend to consider problems as unique, not accounting for similar cases. That is, expert estimations rely on the “inside” view, which only takes the structure and the impediments of the specific case into account. An “outside” view, on the other hand, takes *distributional* information of similar cases into account (Kahneman and Lovallo, 1993).

Studies that investigate what factors relate to execution time offer useful insights to take a more “outside” view to time estimates. Lanigan, (1994) showed that task effort is a function of the *nature* of the task itself and the number of people working on it. Kakimoto et al., (2018) showed that maximum team size to estimate effort of a project is effective and robust to perturbations when the error rate is equal or less than 50%. In software engineering, different studies relate the size of software, captured by number of lines of code, function points, or number of files, to the execution time or development effort (Albrecht and Gaffney, 1983; Boehm, 1984; Symons, 1991).

In this paper, we connect the previous insights with the domain of Engineering Design, testing the overall hypothesis that task execution time can be predicted, to some extent, from the properties of the networked structure of the project.

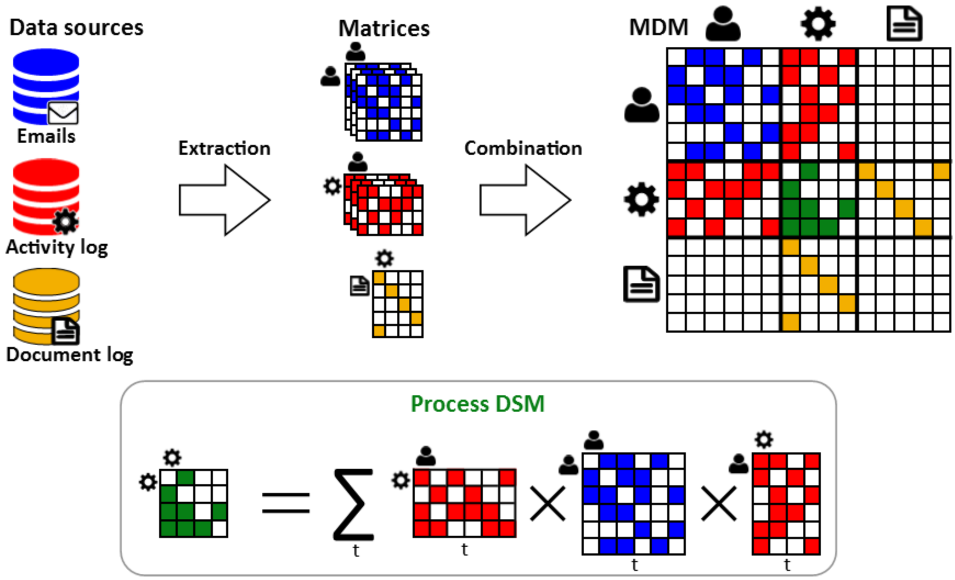


Figure 1: Process of data extraction and combination to build the Multiple Domain Matrix (MDM) used to understand task execution time in relation to project structure.

3 Data and Methods

3.1 Data

The data used in this paper refers to a large-scale design project of a biomass power plant conducted by a multi-project Scandinavian company (Parraguez et al., 2015). Three different data sources are available:

- An activity log, which records the activities performed by the company's personnel throughout the duration of the design process. The activity log describes the relations between 100+ people and ~150 unique activities. Each activity is identified by an activity code assigned by the software that the company uses to manage the project.

- A document log, which contains metadata for the 3000+ documents created during the design process. The metadata include information about document creation and last modification dates, external companies involved in the document editing process, and the code of the activity to which each document is related.

- The complete email exchange between all the people involved in the project (employees, suppliers, external consultants, etc.). The complete email archive amounts to ~54000 emails.

3.2 Methodology to build the MDM automatically from data sources

In order to understand a design project in relation to its multilayer network structure, we need to extract the fundamental networks (matrices) from the data sources, in a way that makes them combinable into one Multiple Domain Matrix (MDM) (Figure 1). From the document log, we extract a matrix that maps each document to the activity it refers to. From the activity log, we extract a series of monthly bipartite networks, also known as Domain Mapping Matrices (DMM), that represent the assignment of people to activities, connecting each person to the activities performed in one month. Similarly, from the email archive, we extract a series of directed networks that connect the company's employees based on the monthly email conversation. As the design process under analysis is closer to a Systems Engineering process rather than an agile one, monthly aggregation is appropriate. We tried other more refined aggregations, such as weekly, but the results remained unchanged.

The activity network that describes the information dependencies between the activities that compose the process is obtained by applying relational algebra for networks. Let PA_t *be the matrix describing the assignment of people to activities at time t, and* PP_t *the communication between people, as captured by the email communication, at time t, the activity network at time t is computed with the following formula:* $AA_t = PA_t^T \cdot PP_t \cdot PA_t$. The final activity network for the MDM is computed by aggregating (summation) all the snapshots AA_t into a single one. The matrices extracted as described above are then aggregated and combined to form the MDM (Figure 1). Considering the evolution over time for *PP* and *PA* is important to avoid an unrealistic process DSM that is too dense, where each activity may be connected to nearly any other (Figure 2).

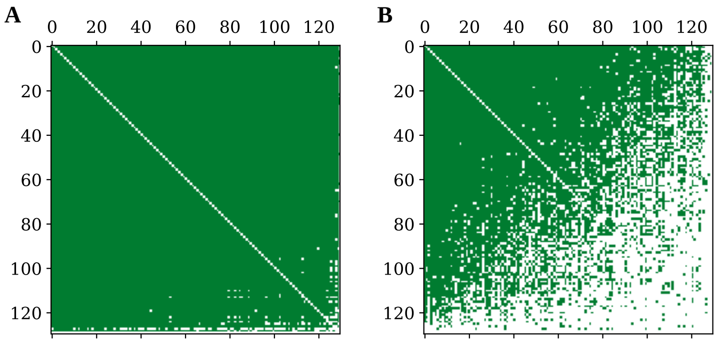


Figure 2: Comparison between the process Design Structure Matrix (DSM) obtained by aggregating all the temporal information into one single snapshot (A) and by using monthly snapshots (B).

3.3 Modelling

In this paper, we focus on understanding task completion time in relation to the multilayer structure of the project. Guided by the insights discussed in the literature review and in accordance with our hypothesis that structural properties of the project can predict, to a certain extent, completion time, we extract the variables of interest from the MDM.

As the completion time for the tasks is expressed in number of days, thus is a positive integer, we use models of the following form:

$$\log(y_i) \sim \alpha + \beta X_i \qquad (1)$$

Where y_i is the completion time for the i-th activity, α is a constant term, X_i is the vector of explanatory variables, and β its relative vector of coefficients. To fit the model we use the ordinary least squares method (OLS) with robust standard errors to account for possible heteroscedasticity. The logarithm transformation of the completion time is useful to reduce the skewness of the distribution. To evaluate the goodness of the models we use the following measures: the R^2, the adjusted R^2, the Akaike Information Criterion (*AIC*), the Bayesian Information Criterion (*BIC*), and the root mean square error (*RMSE*). For R^2 and adjusted R^2, the higher the better; for the other measures, the lower the better. Finally, to check for multicollinearity, we computed the condition number and the variance inflation factors (VIFs). In the following, we describe and discuss the variables that we use in our modelling approach to explain task completion times. The dependent variable, i.e. the variable that we seek to explain using structural properties of the MDM, is the activity execution time. We use activity and document logs to compute the completion time for each activity. As the activity log has data on a daily granularity, we count the number of days elapsed between the first and last time a person worked on an activity or a document connected to it. To account for the *size* of each activity, we compute the number of documents connected to it (#Documents). In the MDM, this corresponds to the degree of the activities in the DMM activity-document (see Figure 1). As each document deals with a set of functional requirements, the number of documents can be interpreted as an approximate measure of the functional requirements of an activity. Furthermore, the number of documents can give a first estimate of the workload of the teams involved (Piccolo et al., 2017). We expect a positive relation between the number of documents and completion time.

For each activity, we compute the number of people (#People) allocated to it as the degree of the activities in the DMM activity-people (see Figure 1). This DMM proved to be highly relevant to understand the role of people in the robustness of a design process (Piccolo et al., 2018). The number of people connected to an activity can be interpreted as an approximation of the workforce needed by the activity. In addition, activities with high number of people assigned to them can be more error prone (Piccolo et al., 2018); thus, we expect a positive relation between the number of people connected to an activity and its completion time.

We account for the structure of the activity network and the amount of information dependencies affecting each activity by computing a set of measures: 1) the degree of each activity (#Activities), i.e. the number of ingoing and outgoing edges; 2) the indegree, i.e. the number of ingoing edges and quantifies the dependency of an activity from the preceding ones; 3) the outdegree, i.e. the number of outgoing edges and quantifies the influence of an activity on the following ones; 4) the product of indegree and outdegree, here termed criticality, which accounts for a synergistic relation between in- and outdegree. We expect a positive relation between these structural properties and the completion time.

Finally, we compute the number of external companies involved in each activity as a possible confounder for the measures computed above. The rationale for including this confounder is that a higher number of external companies involved in an activity could produce more difficulties in coordination and thus, increase overall completion time.

All variables, before the statistical modelling, were normalised by removing their averages and dividing them by their standard deviations. Table 1 shows the correlation between the explanatory variables. We note that the correlation between degree, indegree, outdegree, and criticality is very high (almost perfect correlation). Thus, we present only the models with #Activities, without the other correlated variables to avoid inconsistencies due to multicollinearity. Interpretation for the other variables is the same as for the degree.

Table 1. Correlations between explanatory variables. High correlations ($r \geq 0.7$) highlighted.

	2	3	4	5	6	7
1. # Documents	0.45	0.38	0.37	0.35	0.39	0.4
2. # People		0.36	0.7	0.7	0.71	0.78
3. # Companies			0.42	0.45	0.41	0.47
4. # Activities				0.98	0.98	0.95
5. Indegree					0.95	0.96
6. Outdegree						0.96
7. Criticality						

4 Results

We present the results of our analysis in Table 2. First, we develop a baseline model that accounts only for the effect of the number of documents, the number of people, the number of activities, and the number of external companies involved. All the terms are positive and significant, with the exception of the number of people and the amount of companies. We develop a second model to account for the possibility of non-linearity in the number of people and the activities' degree. The complete model represents an improvement over the baseline with an increase of ~25% for the explained variance (R^2 and Adjusted R^2). The coefficients confirm the expectations of positive relations between the number of people, documents, activities, and completion time.

The number of people and activities are associated non-linearly and monotonically with completion time (see Figure 3 for a visualisation of the relations). Finally, observing that the number of external companies is not significant, we remove it obtaining a reduced model that has the same explanatory power as the previous one. The coefficients remain significant, describing the same positive associations of the variables with the completion time. Our models do not suffer of multicollinearity, as confirmed by the Variance Inflation Factors (VIFs) and condition numbers smaller than 10.

Table 2. Regression table. Dependent variable: execution time

Coefficients	Baseline	Complete	Reduced
Constant	5.49*** (0.13)	6.10*** (0.19)	6.12*** (0.19)
log(#Documents)	0.56*** (0.14)	0.60*** (0.14)	0.54*** (0.13)
#People	0.14 (0.13)	1.02*** (0.25)	1.02*** (0.25)
#People²		-0.38*** (0.10)	-0.37*** (0.10)
#Activities	1.00*** (0.20)	0.36 (0.25)	0.31 (0.24)
#Activities²		-0.24* (0.11)	-0.26* (0.11)
#Companies	-0.16 (0.10)	-0.15 (0.09)	
R^2	0.43	0.52	0.52
Adjusted R2	0.41	0.50	0.50
AIC	493.85	474.36	473.40
BIC	508.19	494.43	490.60
RMSE	7.72	6.95	7.59
#Observations	130	130	130

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$ Standard errors in parentheses

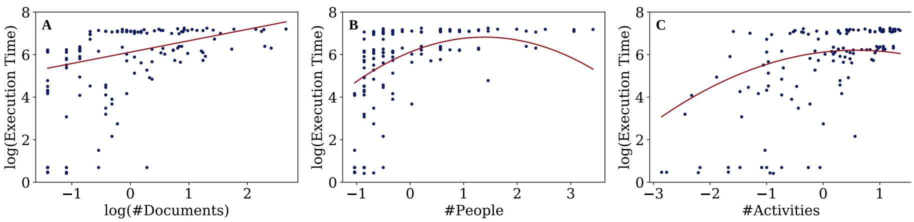


Figure 3: Relations between task execution time and #Documents, #People, and #Activities. The negative numbers are due to variable standardisation.

5 Discussion

Estimating task execution time is an important activity for planning and managing engineering projects, as many scheduling tools require task completion time estimates as one input variable. However, time estimates are often too optimistic because of cognitive biases that prevent experts to realise and consider the many factors influencing task execution. Here, we proposed to understand execution time in relation to the multilayer structure of a project through the use of a Multiple Domain Matrix (MDM). Differently from traditional approaches that rely on interviews, we developed a method to build the MDM automatically from three data sources: email communications, activity logs, and document logs.

While we analysed only one project and the specific value of the regression coefficients pertain only to this case, our analysis produced results in line with current practice in software engineering and insights that we believe are useful to improve the practice of time estimation and project management. We discuss them in the following. Our modelling strategy shows that the task completion time can be modelled as a function of the number of documents produced in the context of the task (task size), the number of people allocated to it (resource allocation), and the number of interfaces with the other tasks (task interfaces).

Task size: The number of documents, here, is a proxy for the *size* of a task, as the lines of code or the number of function points are in software development (Albrecht and Gaffney, 1983; Boehm, 1984; Symons, 1991). The positive relation between the number of documents and execution time (see Figure 3A) shows that "task sizing" can be useful also outside software engineering. We found that the logarithm of the number of documents performs better than the crude number, which means that a perfect estimation of the size is not necessary and a measure of the order of magnitude would perform well. Understanding which measures of task size are the most suitable for engineering design is a topic for future research and we suspect that a measure derived from the functional requirements, as it happens in software engineering (ISO, 2007), can be a good starting point.

Resource allocation: We have also found a positive relationship between the number of people assigned to an activity and its execution time. In Figure 3B, it is clear that the relation is monotonic. The quadratic curve starts decreasing after ~90.5% of data points and does not represent a good fit anymore. The positive relation between the number of people allocated to an activity and its completion time shows that the amount of people assigned to a task should be used to make time estimations as tasks with higher number of people require more time. This is especially important as it has been documented, under the name *team scaling fallacy*, that underestimation of completion time increases as team size increases (Staats et al., 2012). Furthermore, activities with a high number of people assigned to them are more important for process robustness as errors or changes originating through such tasks can spread faster and affect more activities (Piccolo et al., 2018).

Task interfaces: The number of interfaces an activity has with other activities is also positively associated with the completion time. The relation is monotonic and no turning point is observed (see Figure 3C). Thus, in case of the relation between completion time and number of interfaces, we do not find a curvilinear relation, as for example claimed in Gokpinar et al., (2010) for the relation between the number of interfaces of a subsystem and the number of defects. We also found that the number of interfaces in input (indegree) has almost the same explanatory power as the total count of interfaces. This means that the completion time is in direct relation with the number of inputs that a task has to integrate. The relation between the degree and completion time reminds us of the importance of integrative activities during error propagation processes (Braha and Bar-Yam, 2007; Piccolo et al., 2018).

With a measure of activity size, people assigned to activities, and number of interfaces we were able to explain 50% of variance in the completion time. We argue that these measures, such as the number of people allocated to a task, are readily available or can be estimated during the planning stage of a project. The number of interfaces per activity can be obtained by building the DSM for process sequencing. The measure of activity size could be estimated from the amount of functional requirements. One could be tempted to explain more variance by adding more variables to the models. While there are definitely many more factors that can affect task completion time, it is worthy to remember that the use of irrelevant information hinders good time estimates (Halkjelsvik and Jørgensen, 2018). We believe that the process of data analysis and the measures used here can be used to support experts in making better estimates, while helping them to take a more outside view (Kahneman and Lovallo, 1993). Studying how to integrate these and other metrics as well as the process of data analysis into the practice of project management is a topic for future research.

6 Conclusions

Estimating task completion time is difficult and often results in underestimates due to optimism bias and other human and situational biases and a lack of meaningful information on which to base the estimates. To provide a ground for better estimates, this paper combined and analysed multiple data sources from a large-scale design project, showing that task completion time relates to the structure of the project as captured by a Multiple Domain Matrix (MDM). Statistical analyses showed that task execution time correlates positively with the *size* of the task, the number of *interfaces* with other tasks, and the number of *people* allocated to the task. In our case, we were able to explain 50% of the variance. We discussed implications of the findings and gave pointers on how the three metrics used can be made available to managers during the planning stage of a project. Moreover, this study also showed a possible use of historical data to inform future decision-making.

References

Albrecht, A.J., Gaffney, J.E., 1983. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. IEEE Trans. Softw. Eng. SE-9, 639–648. https://doi.org/10.1109/TSE.1983.235271

Bashir, H.A., Thomson, V., 2001. Models for estimating design effort and time. Des. Stud. 22, 141–155. https://doi.org/10.1016/S0142-694X(00)00014-4

Boehm, B.W., 1984. Software Engineering Economics. IEEE Trans. Softw. Eng. SE-10, 4–21. https://doi.org/10.1109/TSE.1984.5010193

Braha, D., Bar-Yam, Y., 2007. The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results. Manage. Sci. 53, 1127–1145. https://doi.org/10.1287/mnsc.1060.0617

Eppinger, S.D., Browning, T.R., 2012. Design Structure Matrix Methods and Applications. The MIT Press.

Flyvbjerg, B., 2006. From Nobel Prize to Project Management: Getting Risks Right. Proj. Manag. J. 37, 18–19.

Gokpinar, B., Hopp, W.J., Iravani, S.M.R., 2010. The Impact of Misalignment of Organizational Structure and Product Architecture on Quality in Complex Product Development. Manage. Sci. 56, 468–484. https://doi.org/10.1287/mnsc.1090.1117

Halkjelsvik, T., Jørgensen, M., 2018. Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-74953-2

ISO, 2007. Information technology – Software measurement – Functional size measurement. Part 1: Definition of concepts, International Standard ISO/IEC 14143-1.

Jørgensen, M., 2004. A review of studies on expert estimation of software development effort. J. Syst. Softw. 70, 37–60. https://doi.org/10.1016/S0164-1212(02)00156-5

Kahneman, D., Lovallo, D., 1993. Timid Choices and Bold Forecasts: A Cognitive Perspective on Risk Taking. Manage. Sci. 39, 17–31. https://doi.org/10.1287/mnsc.39.1.17

Kakimoto, T., Tsunoda, M., Monden, A., 2018. Should Duration and Team Size Be Used for Effort Estimation?, in: Lee, R. (Ed.), Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. Springer International Publishing, Cham, pp. 91–105. https://doi.org/10.1007/978-3-319-62048-0_7

Lanigan, M.J., 1994. Task estimating: completion time versus team size. Eng. Manag. J. 4, 212. https://doi.org/10.1049/em:19940508

Murmann, P.A., 1994. Expected Development Time Reductions in the German Mechanical Engineering Industry. J. Prod. Innov. Manag. 11, 236–252. https://doi.org/10.1111/1540-5885.1130236

O'Donovan, B., Eckert, C., Clarkson, J., Browning, T.R., 2005. Design planning and modelling, in: Design Process Improvement. Springer, pp. 60–87.

Parraguez, P., Eppinger, S.D., Maier, A.M., 2015. Information Flow Through Stages of Complex Engineering Design Projects: A Dynamic Network Analysis Approach. IEEE Trans. Eng. Manag. 62, 604–617. https://doi.org/10.1109/TEM.2015.2469680

Piccolo, S.A., Lehmann, S., Maier, A., 2018. Design process robustness: a bipartite network analysis reveals the central importance of people. Des. Sci. 4, e1. https://doi.org/10.1017/dsj.2017.32

Piccolo, S., Lehmann, S., Maier, A., 2017. Using data- and network science to reveal iterations and phase-transitions in the design process, in: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 2: Design Processes | Design Organisation and Management. Design Society, pp. 11–21.

Project Management Institute, 2017. A guide to the project management body of knowledge (PMBOK ® guide), Project Management Journal.

Srinivasan, K., Fisher, D., 1995. Machine learning approaches to estimating software development effort. IEEE Trans. Softw. Eng. 21, 126–137. https://doi.org/10.1109/32.345828

Staats, B.R., Milkman, K.L., Fox, C.R., 2012. The team scaling fallacy: Underestimating the declining efficiency of larger teams. Organ. Behav. Hum. Decis. Process. 118, 132–142. https://doi.org/10.1016/j.obhdp.2012.03.002

Symons, C.R., 1991. Software Sizing and Estimating: Mk II FPA (Function Point Analysis). John Wiley & Sons, Inc., New York, NY, USA.

Thamhain, H.J., Wilemon, D.L., 1986. Criteria for controlling projects according to plan. Proj. Manag. J. 17, 75–81.

Contact: Sebastiano A. Piccolo, Technical University of Denmark (DTU), Department of Management Engineering, Produktionstorvet, Building 424 DK-2800, Kongens Lyngby, Denmark, sebpi@dtu.dk

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Modelling of Digital Extended Enterprise

Antti J. Pulkkinen[1], Ville V. Vainio[1], Simo-Pekka Leino[2], Juha-Pekka Anttila[2]

[1]Tampere University of Technology, Finland
[2]VTT Technical Research Centre of Finland

Abstract: The concept of Digital Extended Enterprise is presented as a form of highly competitive manufacturing network. The maturity model of digital extended enterprise is defined as a method for the management of the development of manufacturing networks. The research methods of this paper are literature studies in many topics as well as action research with five industrial companies. Our findings support the use of the digital maturity model in the qualitative assessment of development projects. The quantified business indicators can be used for evaluating the attained benefits, while the maturity model is used for assessing the development efforts and steps taken. A dependency matrix for studying the maturity model is defined. We address the oversimplification problem of maturity modelling and suggest the use of dependency matrix to alleviate the oversimplification problem, to improve the maturity model and to enhance the knowledge on the development of digital extended enterprise.

Keywords: Extended enterprise, Digitalization, Modelling, Maturity, Dependency

1 Introduction

In this article we briefly present how to assess the capabilities of a supply network by using digital extended enterprise concept and maturity models. In the use of the maturity model we realized that the model itself contains overlapping and interrelated areas. The further development of the maturity model requires improved understanding of these matters and the understanding of the maturity itself; the interrelations of different topics within the maturity model may present new knowledge for the development of supply networks.

The research project this article is a part of has taken place along with a concurrent industrial project for more than two years. In the project, an OEM supplier of mineral aggregates handling machinery and a selected few of its sub-suppliers have developed networked strategies, operations, digitalization, processes, business indicators, etc. The OEM supplier has selected these development activities based on its strategy as well as on the problems encountered in its supply network. The partners have developed the internal operations management and visualization, manufacturing and engineering processes, personnel capabilities, business indicators, IT-systems, etc. and the OEM supplier has collaborated with a hands-on attitude. The industrial development project has provided the material and the phenomena of developing an extended enterprise to the research.

1.1 Digital Extended Enterprise

During the last couple of decades, the organization of industrial activities has evolved towards manufacturing networks. It is an obvious path of evolution related to specialization and the focusing of core competence. For the outsourcing of operations and the development of procurement, several methods have been introduced. Most notable of them is the Krajlic's matrix (Krajlic,1983), which defines four categories of items by relating profit against risk and consequently suggests respective characteristics of supply-procurement situation.

The definition of suppliers' characteristics is a challenge, which remains even after original equipment manufacturer (OEM) has defined the characteristics of items and classified the existing items according to the characterization. For example, Momme (2002) suggested a framework for outsourcing manufacturing and Gelderman and Semejin (2006) a methodology for procurement portfolio management for selecting the most suitable manufacturing suppliers.

However, it is not always an option to start selecting new suppliers due to reasons of economy, time, logistics or availability. Instead of supplier selection, the methods for the collaborative development of manufacturing networks and OEM-supplier relationship should be applied. In this situation, it is necessary to assess the existing suppliers and relations with them (Momme 2002).

The concept of Extended Enterprise (EE) integrates business strategies and operational modes, engineering and manufacturing processes. The objectives of EE are persistent and distinctive operational transformations that will lead to a set of benefits beyond the results of traditional technical or business process re-engineering cases. (Browne et al. 1995) Similar concepts are collaborative supply chain (Simatupang, & Sridharan, 2002) and collaborative networked organizations (Camarinha-Matos et al. 2009) as well as industrial platforms and ecosystems (Gawer,& Cusumano 2014).

The integration and communication aspects of an extended enterprise have been recognized for long time (Browne et al. 1995). The goal of an extended enterprise is the total optimization and competitiveness of the whole network. The role of digitalization is to enable the integration and transformation when an extended enterprise as a whole can utilize the traditional production development methodologies, such as Lean, within and over the organizations of a network (Xu 2015, Burton and Bodeur 2002). Thus, we consider the contemporary version of an optimal manufacturing network a digital extended enterprise (DEXTER).

1.2 Maturity Models

The word maturity is defined as "the state of being fully grown or developed" [Hornby, A. S., Wehmeier, S., McIntosh, C., Turnbull, J., & Ashby, M. (2005). *Oxford advanced learner's dictionary of current English* (7th ed.). Oxford University Press.]. In the context of a multistage model, maturity implies the evolution of a specific subject from an initial to a fully developed stage.

Capability Maturity Model® (CMM®) was developed by Carnegie Mellon with an original intention "…to characterize the capabilities of software-development organizations" (Humphrey 1988, p. 73). Later Software Engineering Institute has developed a set of Capability Maturity Model Integration (CMMI) reference models for a variety of purposes, such as acquisition, development and services (CMMI 2010).

For the modelling of manufacturing capability, Britton et al. (2007) used the breakdown structure of Production System Design. They modeled the part-of structure of functional requirements (FRs) with performance metrics (M) and plausible physical solutions (PSs) and analyzed the model with mind maps and matrices (see Fig 2.). The purpose was to define relevant questions for manufacturing assessment.

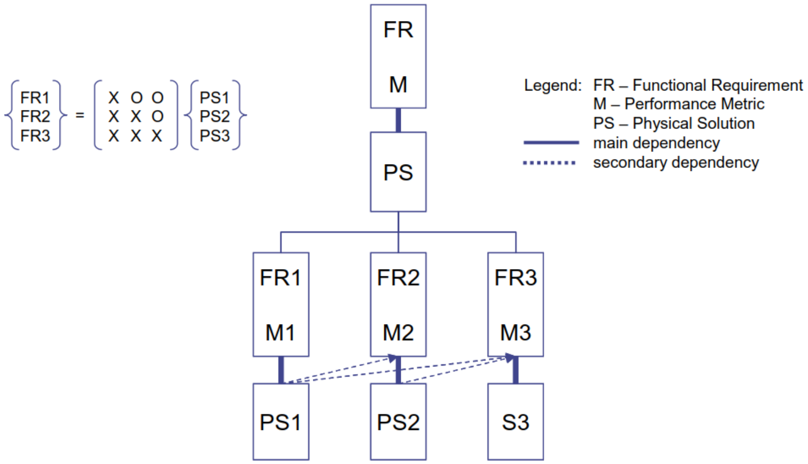


Figure 1. Illustrative decomposition of Production System Design (Britton et al. 2007, p. 3)

Eventually, Britton et al. (2007) structured the assessment questions in the form of maturity assessment statements or questions concerning the performance metrics and solutions of different level of decomposition. The order of maturity assessment followed the dependencies of the current solutions.

Maturity models for supply chain assessment and development has been the focus of many researchers and consultants. Lahti et. al (2009) studied six models that were published during the years 1999 and 2007. All the studied models were “based on years of supply chain benchmarking experience… tested, tried and proved ” (Lahti et al. 2009, p. 666). The models had common themes, but their sources and data collection varied. The models focused on the extended connectivity and integration, collaboration, evolution from dysfunctional to strategic partnership. Based on the models, Lahti et al. (2009) developed and tested their own supply chain maturity model that comprised of four maturity stages and key performance areas (KPA): plan, source, make, deliver and overall.

Done (2011) developed also another supply chain maturity model. He focused on the relation of performance and maturity and emphasized the downstream and upstream planning as the means of collaborative and integrative practices. Other KPAs in his model were sourcing, making, delivering, NPD as well as upstream and downstream return maturity. With the analysis of large sample respondents to an online survey Done claimed that the framework constitutes an appropriate model for the concept of supply chain maturity. He also found out that there is a “strong statistical support regarding the significant impact of supply chain maturity dimensions on multiple objective performance measures”. Done (2011, p. 24).

Leino et al. (2017a) have defined the model of digimaturity as an aid for understanding and structuring the concept of digitalization. An organization can assess its capabilities in six dimensions of the associated tool: strategy, business model, customer interface, organization and processes, people and culture, and information technology. Leino and Anttila (2017b) suggested that a new version of the model could be defined for the assessment of manufacturing industry.

Common for all maturity models are the levels of maturity, which is a standardized way to measure the situation of different KPAs or topic areas of the model. Humphrey (1988) defined the five process maturity levels: initial, repeatable, defined, managed and optimizing. CMMI (2010) separates the concepts of capability and maturity and therefore defines the subtler levels from 0 to 5. In addition, Britton et al. (2007) used levels from 0 to 5. The studied supply chain maturity models (Lahti et al. 2009, Done 2011) had maturity levels ranging from 4 to 5.

"A Capability Maturity Model® is a simplified representation of the world" (CMMI 2010, p. 5). While evaluating the maturity level of a subject, there is a risk of oversimplifying reality and not having empirical basis. In addition, the common lack of a linear sequence of stages in organizational life must be acknowledged. (Poeppelbuss et al. 2011)

1.3 Design Structure Matrix

Design or Dependency Structure Matrices (DSMs) have been used to model the relations and dependencies of objects, such as complex systems (e.g. electronic systems, buildings, aircraft and automobiles), business, manufacturing and engineering processes and organizations (Eppinger & Browning 2012, Browning 2016). The most known use of the method is to manage the structural complexity of a system experienced within product development and engineering (Eppinger 1994, Lindemann et al. 2009).

Eppinger and Salminen (2001), Browning (2001) and Bongulielmi et al. (2001) presented the concept of using matrices to represent the relations between the objects of different domains. Malmqvist (2002) considered the relations of the objects of one domain as intra-domain and the relations of the objects of two domains as inter-domain relations. Maurer (2007) coined the term Multi Domain Matrix (MDM).

Bongulielmi et al. (2001) and (Nummela) 2006 presented the variants of MDM and DSM to represent knowledge on the dependencies of product family for product configuration. Even the relations of functional requirements and physical solutions can be represented with an MDM as can be seen in the top left corner of Fig 1.

Despite the large variety of applications, DSMs rarely cover dynamic dependencies or system maturities. For example, Eppinger & Browning (2012) report only two case examples where change in time was incorporated in DSM or MDM representation. For example, duration, information maturity and probability of change are "additional attributes… that are not usually shown explicitly in a DSM" (Eppinger & Browning 2012, p. 140). In addition, Bongulielmi et al. (2001) stated that conditional dependencies cannot be represented in K- & V-matrix method.

2 Modelling digital extended enterprise

We developed a model of Digital Extended Enterprise (DEXTER), which comprises of sub-models: Dexter definitions, Dexter concept model and Dexter maturity model. We derived the concept model topics from the definitions from literature (see 1.1). The definitions included following domains: extended enterprise strategy, organizations, processes, supply network structure, business indicators, change management, products and services, IT as an enabler, information flows and interfaces.

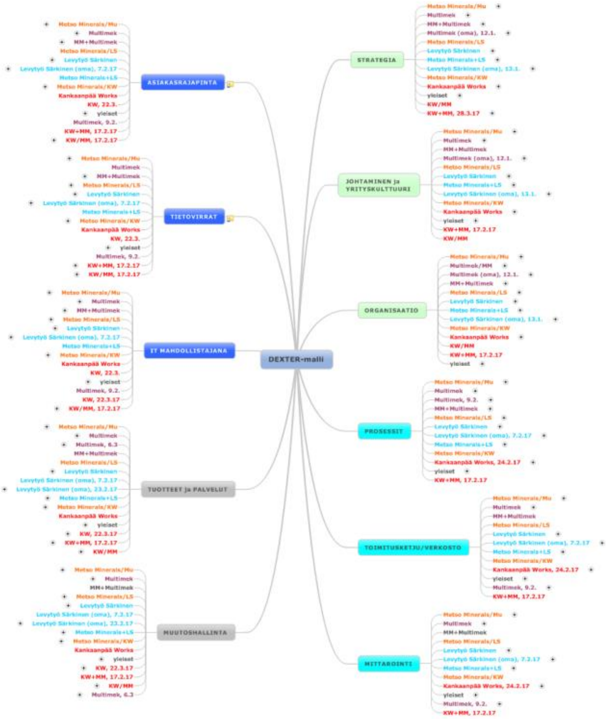


Figure 2. Dexter concept model (partial)

The concept model is a mind map diagram composed of all the topics, which the case companies found to be important in relation to the effectiveness of an Extended Enterprise network. We tested and further developed the model in co-operation with five manufacturing industry companies. As the figure above indicates, we documented implications of practical aspects in each domain from each interview in the concept model. This served the development of the extended enterprise: we recognized the relevant domains in practice and we defined pilot projects based on the concept model.

However, conducting interviews, documenting and assessing was tedious and time consuming. Moreover, the assessment was based on on collecting similar opinions of the current state. Therefore, it became evident that collaboration within an extended enterprise needs a faster and more structured approach to discover the statuses and development targets within organizations.

2.1 DEXTER Maturity Model

For improved understanding of the state of an extended enterprise, it is necessary to have a structured method for assessing the state of the topics in the domains characterizing the

EE. Moreover, in an extended enterprise one must conduct the assessment with uniform measures and methods to collect equivalent information and to make valid comparisons of the states of topics in time and in the different organizations.

In the beginning of this research project, we did not aim to model maturity and did not seek any ready-to-use maturity models for our purposes, which led us to develop a suitable one. For example, we did not refer to such a structure and a method as Production System Development method like Britton et al. (2007) did in their maturity model. Rather, we focused on the literature on digital extended enterprise and the basis of the maturity model was the Dexter concept model. We further developed the maturity model with iterations between researchers and practitioners, mainly the procurement development team of OEM in the project. We analyzed the topics of the concept model, findings from the interviews and formed categorized sets of questions, which resulted in a maturity level questionnaire.

For the current version of the questionnaire, we chose the following six main domains: A) Strategy, B) Business Model, C) Processes, D) Performance indicators, E) Interfaces and F) Information flow. For the convenience of the respondents, we divided the categories into three forms: A+B, C and D+E+F. Answering each form took from twenty minutes to one hour. In total, there were 76 questions to be answered, of which 69 were multiple choice questions with five options. Generally, these five answers represent the maturity level on a scale of zero to four, four being the most developed or mature state. An example of a part of the questionnaire is in Figure 3.

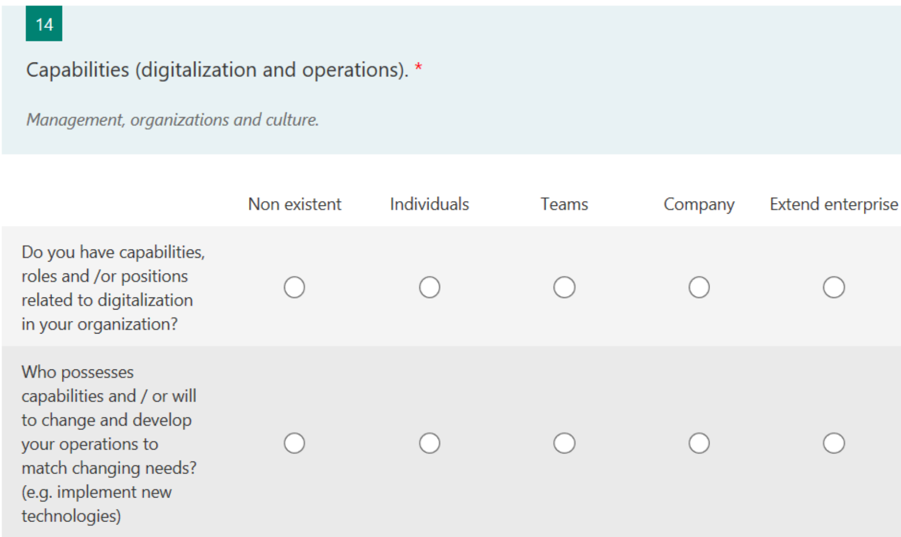


Figure 3. An example of Dexter maturity question (domain: Business model)

Every company answered each form of the questionnaire twice. Once considering the state of spring 2018 and once as to what the state was in the first quarter of the year 2016, i.e. in the beginning of the research and development project. The level of maturity was not always explicit in a question (see Figure 3), but each option was coded to match a certain level of maturity in the analysis of answers. For example, the answer "*Nonexistent*"to the statement "*Do you have capabilities, roles, and/or positions related to digitalization in your organization*" would correspond to level 0 in the topic 14a of the domain business model.

2.2 DEXTER Maturity Model Dependencies

There is no theory to put the different domains of DEXTER maturity model nor the questions of the model in order. We cannot predict a situation within a manufacturing network prior to a study. However, our study suggests that a digital extended enterprise is a combination of different domains, which are interrelated in practice. Therefore, adopting a matrix representation of the current version of the maturity model appeared potential for the development of the model itself further and we defined an MDM based on the maturity model.

We marked the domains and the questions of the domain in first rows and columns of a Dexter Maturity Dependency Matrix (DM^2) and adhered to the structure of the maturity model in the DM^2. The DM^2 cells represented the relations of maturity levels within the topics of domains and between the topics of different domains. For example, we marked that level 3, i.e. "*Company*" of question / statement 14a **requires** level 2 of question/ statement 14b, i.e. "*Teams*" (see Figure 3). On the other hand, we may indicate that the *Teams* with the *capabilities, roles, and/or positions reated to digitalization* **enable** some other level in the domain *information flow*, with an inter-domain relationship in the DM^2.

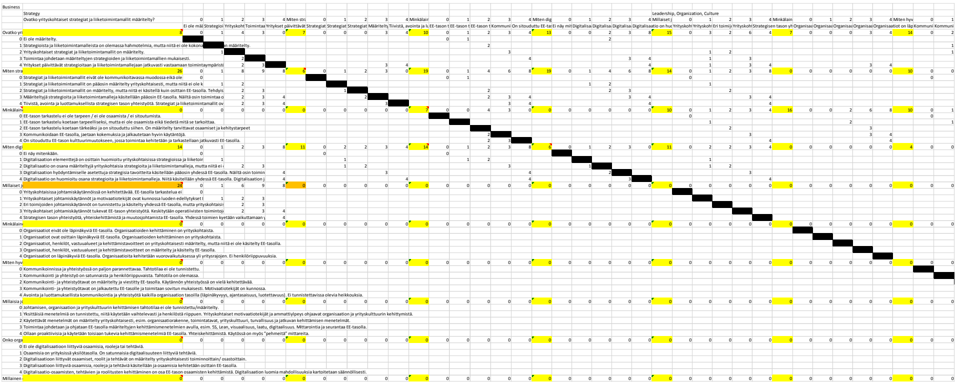


Figure 4. Partial DM^2

Furthermore, it is possible to indicate the level of maturity required in the cells representing relations between different topics of maturity with numbers. Instead, we may mark different kind of relation dependencies, such as "precedes X", "enables X", "enhances X", "removes X", "removes the need of X" or "depends on X", with different numbers.

3 Findings & Conclusions

We compared the findings of our maturity assessment to the business indicators that the OEM procurement development team provided to us. The quantified indicators, such as the improved quality and reliability of deliveries, and decreased losses due to obsolete items, presented the success of suppliers' development projects.

We studied the capabilities and results both before and after the development actions to find qualified results of the actions. As expected, persons with different expertise and experience levels answered the questions with different attitude. However, it was easy to recognize the trends, even though some respondents regarded the *status quo ante* less satisfactory as well as the steps of maturity more moderate than other respondents. In a couple of domains, such as business indicators, we could calculate from the responses a relatively high rise of maturity throughout the extended enterprise. The highest topic specific rise of maturity was 2.4, levels, highest domain specific rise was 1.3 levels and in general, the development average was 0.8 levels. Thus, we were able to recognize the actions in a variety of levels and the effects at a high level.

The current version of DM^2 is only a draft, because it is incomplete and only two researchers have developed the rationale behind the relations of the DM^2. Nevertheless, the existence of relations appears to be rather self-evident, but the number of relations is high. This requires a systematic approach to model the DM^2 and we are currently contemplating it.

Typically, the mind-set of maturity modelling is to move systematically from the one level of maturity to the next one. In the many cases of intra-domain relations this appears to be the case as the level 2 requires the level 1 and enables the level 3. One might assume that the development of digital extended enterprise is straightforward.

However, even with an incomplete model it is possible to start to recognize that the reason of oversimplification of maturity models may be the lack of dependencies between the aspects of maturity. In the model, this means that the dependencies are often inter-domain relations, which indicate the need to traverse from one aspect of development to another to attain the next level of maturity, which validates the finding by Poeppelbuss et al. (2011).

The more rudiment weakness of maturity models is the presumption of independence. Often the maturity models are targeted to wicked problems, such as enterprise networks that "… are ill-formulated, where the information is confusing, where there are many clients and decision makers with conflicting values…" C. West Churchman (1967, p. B-141).

Based on the abovementioned findings it is evident that the DEXTER Maturity model is a viable option for the management and assessment of the progress of development in an extended enterprise. The DM^2 modelling indicates the use of Multi Domain Matrix can provide insight and documentation on the knowledge of the management of extended enterprise development. Furthermore, it may address the dilemma of the oversimplification of wicked problems with maturity model. In addition, the DM2 will be used for the development of the DEXTER Maturity model.

References

Bongulielmi, L., P. Henseler, C. Puls and M. Meier (2001). The K- & V-Matrix Method -an Approach in Analysis and Description of Variant Products. ICED 01: International Conference on Engineering Design, Glasgow; August 21-23, Professional Engineering Publishing Limited; Bury St Edmunds and London; UK.

Britton, G., Huhtala, P. Koho, M., Torvinen, S. (2007) Production System Maturity Model. In the proceedings of 2nd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2007). Toronto, Canada on 22-24 July 2007. Available at (retrieved 6.6.2018): https://s3.amazonaws.com/academia.edu.documents/7556518/CARV2007FullPaperv1.1.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1528725152&Signature=dCpYFYqo9wYYwelu%2FHRGKLhx7aU%3D&response-content-disposition=inline%3B%20filename%3DProduct_System_Maturity_Model.pdf

Browne, J., Sackett, P.J., Wortmann, J.C. (1995) Future manufacturing systems — Towards the extended enterprise, Computers in Industry, Volume 25, Issue 3, pp. 235-254, ISSN 0166-3615, https://doi.org/10.1016/0166-3615(94)00035-O.

Browning, T.R. (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new directions. in IEEE Transactions on Engineering Management, vol. 48, no. 3, pp. 292-306, Aug 2001. doi: 10.1109/17.946528

Browning, T.R. (2016) Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. in IEEE Transactions on Engineering Management, vol. 63, no. 1, pp. 27-52, Feb. 2016. doi: 10.1109/TEM.2015.2491283

Burton, T.T., Boeder, S.M. (2003) The Lean Extended Enterprise – Moving Beyond the Four Walls to Value Stream Excellence. J.Ross Publishing. ISBN-13: 978-1932159127

Camarinha-Matos, L.M., Afsarmanesh, H., Galeano, N., Molina, A. (2009). Collaborative networked organizations – Concepts and practice in manufacturing enterprises, Computers & Industrial Engineering, Volume 57, Issue 1, p.46-60, https://doi.org/10.1016/j.cie.2008.11.024.

Churchman, C.W. (1967) Wicked problems. Management Science, 14(4). https://doi.org/10.1287/mnsc.14.4.B141

CMMI 2010. CMMI for Acquisition, Version 1.3. (Version 1.3, November 2010). Carnegie Mellon University Software Engineering Institute. 2010. Available at (retrieved 6.6.2018): https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15284.pdf

Done, A. Developing supply chain maturity. Working Paper. WP-898. January, 2011. IESE Business School – University of Navarra. Available at (retrieved 15.8.2018): https://www.researchgate.net/publication/241755978_Developing_supply_chain_maturity

Eppinger, S.D., Whitney, D.E., Smith, R.P., Gebala, D.A. (1994). A model-based method for organizing tasks in product development. Research in Engineering Design (1994) Volume 6: Issue 1. pp 1–13. https://doi.org/10.1007/BF01588087

Eppinger, S., Salminen, V. (2001) Mapping of interactions in the product, organization, process architectures, ICED Proceedings, 2001

Eppinger, S.D., Browning, T.R. (2012) Design Structure Matrix Methods and Applications. MIT Press. ISBN-13: 978-0262017527

Gawer, A. and Cusumano, M. A. (2014), Platforms and Innovation. J Prod Innov Manag, 31: 417-433. doi:10.1111/jpim.12105

Gelderman, C.J., Semeijn, J. (2006) Managing the global supply base through purchasing portfolio management, Journal of Purchasing and Supply Management, Volume 12, Issue 4, 2006, pp. 209-217, https://doi.org/10.1016/j.pursup.2006.10.002.

Hornby, A. S., Wehmeier, S., McIntosh, C., Turnbull, J., & Ashby, M. (2005). Oxford advanced learner's dictionary of current English (7th ed.). Oxford University Press

Humphrey, W. S. 1988. Characterizing the software process: a maturity framework. IEEE Software. 5 (2): 73–79. doi:10.1109/52.2014.

Kraljic, P., (1983). Purchasing must become supply management. Harvard Business Review 61 (5), 109–117.

Lahti, M., Shamsuzzoha, A.H.M., Helo, P. Developing a maturity model for Supply Chain Management. In: Int. J. Logistics Systems and Management, Vol. 5, No. 6, 2009.

Leino, S-P., Kuusisto, O., Paasi, J., Tihinen, Mt. (2017a). VTT Model of Digimaturity. In: Towards a new era in manufacturing. Final report of VTT's For Industry spearhead programme. VTT Technology: 288. Available at (retrieved 6.6.2018): http://www.vtt.fi/inf/pdf/technology/2017/T288.pdf

Leino, S-P., Anttila, J-P. (2017b) Digimaturity in Manufacturing Industry. In: Proceedings of the 2nd Annual SMACC Research Seminar 2017 Available at (retrieved 6.6.2018): http://smacc.fi/wp-content/uploads/2017/11/Proceedings-of-the-2nd-Annual-SMACC-reserch-Seminar-2017.pdf#page=31

Lindemann, U., Maurer, M. Braun, T. (2009) Structural Complexity Management – An Approach for the Field of Product Design. Springer-Verlag. ISBN: 978-3-540-87888-9, doi: 10.1007/978-3-540-87889-6

Malmqvist, J.(2001) A Classification of Matrix-based Methods for Product Modeling. In: DS 30: Proceedings of DESIGN 2002, the 7th International Design Conference, Dubrovnik, Available at (retrieved 6.6.2018): https://www.designsociety.org/publication/29562/A+Classification+of+Matrix-based+Methods+for+Product+Modeling

Maurer, M. (2007) Structural Awareness in Complex Product Design. Dissertation. TU München. Available at (retrieved 6.6.2018): https://mediatum.ub.tum.de/node?id=622288

Momme,J. (2002) Framework for outsourcing manufacturing: strategic and operational implications, Computers in Industry, Volume 49, Issue 1, 2002, pp. 59-75, ISSN 0166-3615, https://doi.org/10.1016/S0166-3615(02)00059-3.

Nummela, J. (2006) Integrated configuration knowledge management by configuration matrices - A framework for representing configuration knowledge. Dissertation. Publication number 589, Tampere University of Technology. ISBN: 952-15-1564-3

Poeppelbuss, J., Niehaves, B., Simons, A., Becker, J. (2011). Maturity Models in Information Systems Research: Literature Search and Analysis. Communications of the Association for Information Systems: Vol. 29, Article 27.

Simatupang, T.M., Sridharan, R. (2002) The Collaborative Supply Chain, The International Journal of Logistics Management, Vol. 13 Issue: 1, pp.15-30, https://doi.org/10.1108/09574090210806333

Xu, L.D. (2015). Enterprise Integration and Information Architecture – A Systems Perspective on Industrial Information Integration. CRC Press

Contact: A. J. Pulkkinen, Tampere University of Technology, Department of Technical Sciences, Korkeakoulunkatu 6, PO Box 589, 33101 Tampere, Finland, Phone: +358503289404, e-mail: antti.pulkkinen@tut.fi, URL: http://fi.linkedin.com/in/pulkkinen

Part V: Product Development

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

DSM-Based Methods to Represent Specialization Relationships in a Concept Framework

Yaroslav Menshenin[1], Edward Crawley[2]

[1] Skolkovo Institute of Science and Technology
[2] Massachusetts Institute of Technology

Abstract: DSMs and related matrices are commonly used to represent system decomposition, structure, interaction and function/form assignment. But in conceptual design we must also represent specialization that relates a general thing and a type of that thing. In this paper we propose DSM-based methods to represent specialization relationships that occur in conceptual design. The research questions are: how can we encode in a DSM the information about specialization of a concept's processes and instruments; and how it complements the existing approaches of representing the decomposition relationships. The fundamental utility of the proposed approach is that it facilitates the development of alternative concepts during the conceptual design phase blending the information about specialization and decomposition relationships in united framework. This work also proposes a measure of the formal conceptual similarity between alternative concepts.

Keywords: DSM, concept, model-based conceptual design, specialization

1 Introduction

Specialization relationships play an important role in conceptual design phase, as it narrows down the set of alternative concepts. Our work is motivated by the desire to explore specialization relationships, and to encode them in a matrices-based framework, which would enable the identification of alternative concepts satisfying a highly abstracted function. This would also create an opportunity to use the quantitative measures for estimation of formal conceptual similarity between alternative concepts. The objective of this paper is to develop and present a DSM-based framework to represent specialization relationships that commonly occur between concept's processes and forms – especially during conceptual design phase (Pahl and Beitz, 2007).

The Design Structure Matrix (DSM) developed by Steward (1981) is an effective tool to manage a complex system, enabling the matrix models to capture the different DSM applications (Browning, 2001). DSM has been extended to Domain Mapping Matrix (DMM) (Danilovic and Browning, 2004) and Multiple-Domain Matrix (MDM) (Maurer, 2007; Lindemann, 2008). The former is used to facilitate the mapping between two domains, while the latter allows analyzing the system across multiple domains. Eppinger and Browning highlight hierarchical (vertical) and lateral (horizontal) types of relationships, which are important in system modeling (Eppinger and Browning, 2012). The authors argue that vertical relationships stem from the decomposition, while horizontal relationships stem from "interactions between elements, such as flows of material or information, at the same level" (Eppinger and Browning, 2012). Although DSM was applied to above-mentioned types of relationships, to our knowledge it has not been applied to such relationships as specialization (Dori, 2002; Crawley and Colson, 2007), which are fundamentally different than, for example, decomposition (Chiriac et al., 2011). Thus, there is a research opportunity to explore the specialization relationships with support of DSM-based approaches. The research questions are how can we encode in a DSM the information about specialization between concept's processes and instruments, and how it complements the existing approaches of representing the decomposition relationships. The specific objective of this work is to demonstrate unified framework, which supports conceptual design phase by keeping the information about specialized and decomposed processes and forms. Another specific objective of our work is to demonstrate how this information can be used for quantitative assessment of formal conceptual similarity between alternative concepts.

This paper is organized as follows. In section 2 we discuss the difference between specialization and decomposition and its importance in concept framework. Section 3 demonstrates unified framework that presents specialization and decomposition in a DSM. In section 4 we explain the benefit of having the united framework, which is the ability to measure conceptual similarity between alternative concepts. We provide a summary of our work in section 5.

2 Specialization as a transition in design

Decomposition and specialization are two fundamentally different types of relationships between a concept's entries. The core difference is highlighted in the works of Crawley and of Dori. According to Crawley et al., the decomposition is “the dividing of an entity into smaller pieces or constituents” (Crawley et al., 2015). Dori highlights that specialization is “the relation between a general thing and a type of that thing” (Dori, 2002). These two definitions create a clear distinction that the decomposed process or form is a piece of high-level process or form, while the specialized process or form is a type of a high-level process or form.

The example of process decomposition for the process “moving” (with the implicit instrument “vehicle”) is provided in the paper of Deubzar and Lindemann in Figure 1A (Deubzer and Lindemann, 2009). In contrast, Figure 1B shows the specialization of “moving” into three alternative processes – flying (pushing down on air), floating (pushing down on water) and rolling (pushing down on solid ground). The figures 1A and 1B clearly demonstrate the difference between decomposition, realized by dividing process “moving” into smaller sub-processes “storing (energy)”, “converting (energy)”, “using (energy)”, and specialization, realized by relating general process “moving” to such types of that process as “flying”, “floating”, and “rolling”. We see that decomposition and specialization convey different information and both types of information are important and should be considered during the conceptual design phase.

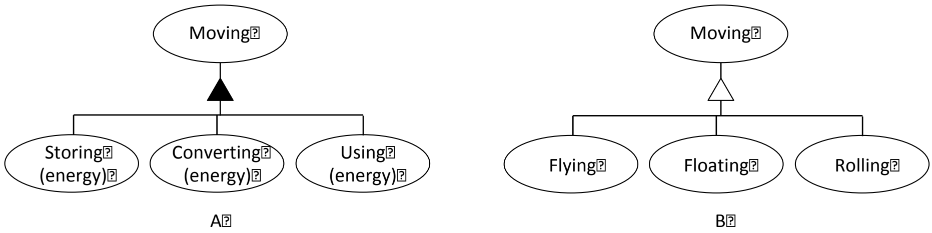


Figure 1. Decomposition (A) and specialization (B) relationships of the process moving

The appearance of the specialization operation in concept is shown in Figure 2A. Following this framework, conceptual design occurs when a solution-neutral process is specialized to a solution-specific process. Using this framework, we extend the “moving” example by showing how the specialization of process and assignment of instrumental form to the specialized processes creates five distinct concepts, illustrated in Figure 2B. As such, the instrumental forms executing the process “flying” are “propeller airplane” or “helicopter”; “floating” process can be performed by “boat”; and “rolling” process can be executed by “car” or “train”. In all examples of Figure 2B the operand is "passengers", as each of the forms serves the purpose to move passengers from one location to another one.

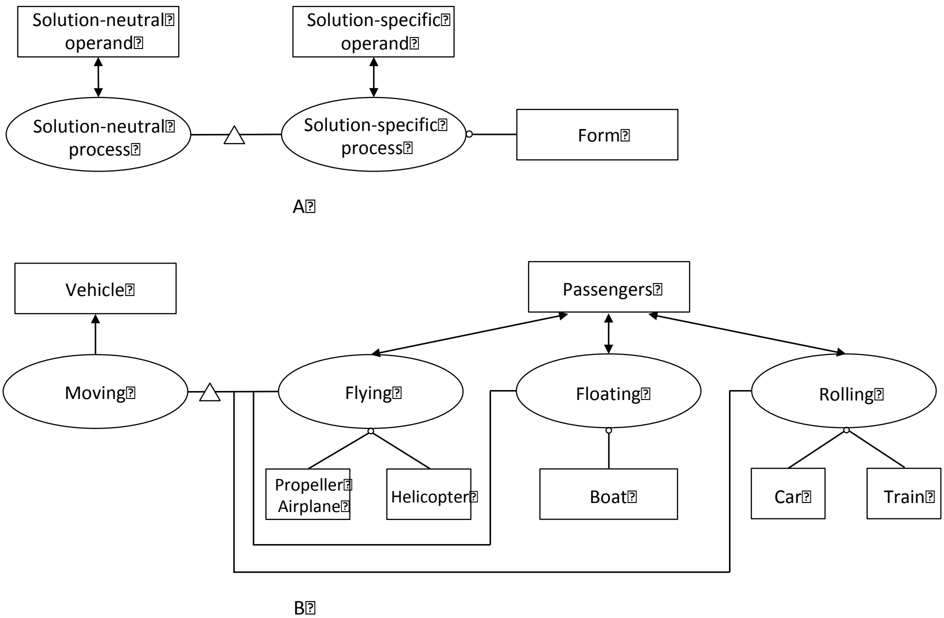


Figure 2. Solution-neutral to solution-specific representation of concept in OPM (A) and example, which specializes the process “moving” to “flying”, “floating”, and “rolling” (B)

3 United framework for representing specialization and decomposition in a DSM

Having established the two complementary operations for specialization and decomposition we seek a way to represent both in a united DSM framework. We further develop the approach by demonstrating the united framework for two alternatives – a “propeller airplane” and a “boat” – in Figure 3. Such united framework contains both types of relationships – specialization of the process/form and the decomposition of specialized processes/forms.

From the exploration of Figure 3 we may see that the form “propeller airplane” is decomposed into three internal forms – “wings”, “propeller”, and “control surfaces”; while the process “flying” is decomposed into three internal processes – “lifting”, “propelling”, and “guiding”. DMM at the lower left corner informs us that “wings” are used for “lifting”, “propeller” is used for “propelling”, and “control surfaces” are used for “guiding” in case of a propeller airplane concept. The same exploration can be done for the second concept – a boat. DSM at the upper left corner maps the information about internal forms to each other. In particular, we can see that both concepts use the same form “propeller” to perform the same process “propelling”. This is denoted by sign “V” at the intersection of “propeller” of “propeller airplane” concept and “propeller” of a “boat” concept. This information sheds light on the idea of conceptual similarity. It should be noted that at the conceptual level we do not distinguish between the “propeller” of the “propeller airplane” and the “propeller” of the “boat.” This is due to the fact that we are intentionally focusing on identification of which form performs which process without going into details about the form itself and its distinctive features.

			Vehicle						Moving					
			Propeller airplane			Boat			Flying			Floating		
			Wings	Propeller	Control surfaces	Hull	Propeller	Rudder	Lifting	Propelling	Guiding	Lifting	Propelling	Guiding
Vehicle	Propeller airplane	Wings	X											
		Propeller		X										
		Control surfaces			X									
	Boat	Hull				X								
		Propeller		V			X							
		Rudder						X						
Moving	Flying	Lifting	V						X					
		Propelling		V						X				
		Guiding			V						X			
	Floating	Lifting				V						X		
		Propelling					V						X	
		Guiding						V						X

Figure 3. United framework of specialization (denoted by white triangle) and decomposition (denoted by black triangle) for two alternatives – a “propeller airplane” and a “boat”

In Figure 4 we present the same information, but for all five concepts mentioned in Figure 2. It is important to note that the decomposed processes for all concepts are the same – “lifting”, “propelling”, and “guiding”, because in Figure 2 we are focusing on the function *moving passengers*. Since the passengers are moved by some kind of vehicle, it is clear that in order to execute any of the processes (“flying”, “floating”, or “rolling”), the decomposed processes “lifting”, “propelling”, and “guiding” must be performed. Let us assume that we need to identify the alternative concepts for a different high-level abstracted function: *moving money*. This example would reveal completely different internal processes, because there are two conceptually different ways to move money: either physically, or electronically. If we try to find the variants to move money physically, we will come up with the same set of alternative concepts as for the *moving passengers* example: for instance, an *airplane*, or a *car*. In this case the internal processes for solution-neutral process *moving* (*money*) would be *lifting*, *propelling*, and *guiding*, because in order to move money we will have to move the vehicle that is used as form. However, since for the client it usually doesn't matter which exactly banknotes he or she uses in the wallet, it looks convenient to move money electronically. This set of concepts would have such internal processes as *depositing*, *e-transferring*, and *withdrawing*. Thus, we may see that not only different processes and forms, but also completely different internal processes and internal forms can be used to achieve the same highly abstracted function.

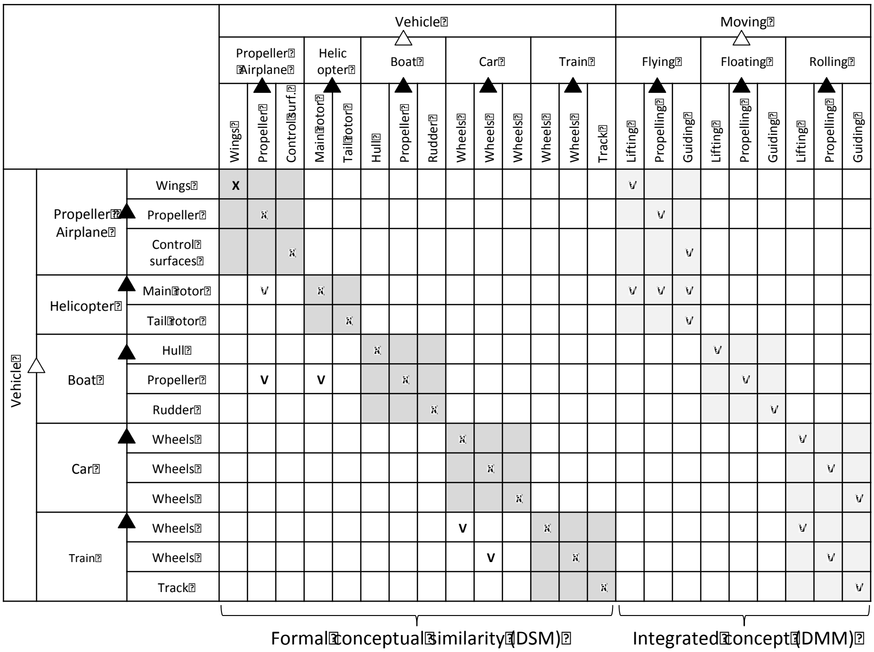


Figure 4. DSM and DMM matrices for five alternatives – “propeller airplane”, “helicopter”, “boat”, “car”, and “train”. Note that the “V” at the intersections of DSM (symmetric matrix) cells denotes information about existence of conceptual similarity between two alternative concepts

From Figure 4 we note the existence of formal conceptual similarity between concepts “propeller airplane” and “helicopter”, “propeller airplane” and “boat”, “helicopter” and “boat”, and “car” and “train”. This information is contained at DSM section of Figure 4, and is denoted by signs “V”. DMM part of the same Figure informs us about the integrated concept, particularly, which exactly internal form is used for which exactly internal process.

4 A benefit of presenting specialization and decomposition in a united DSM – identification of conceptual similarity

One of the benefits is that the framework presented in Figure 4 contains the information about formal conceptual similarity between two alternative concepts. The formal conceptual similarity between each one of the concepts might be measured quantitatively, which is demonstrated in Figure 5. This figure is a DSM-based representation of specialization relationship between form “vehicle” and alternative concepts “propeller airplane”, “helicopter”, “boat”, “car”, and “train”.

		Vehicle				
		Propeller Airplane	Helicopter	Boat	Car	Train
Vehicle	Propeller Airplane					
	Helicopter	1				
	Boat	1	1			
	Car	0	0	0		
	Train	0	0	0	2	

Figure 5. Quantitative measure of formal conceptual similarity for pairs of alternative concepts in DSM (symmetric matrix). The number at the intersection of two concepts indicates how many identical internal forms these alternatives have

Consider such concepts as “propeller airplane” and “helicopter” as an example. The number “1” indicated at the intersection of these two concepts in Figure 5 informs about how many identical internal forms are used between these concepts (in this example, the same internal form is “propeller”). This allows to quantitative measure the formal conceptual similarity between all five alternative concepts.

5 Conclusion

In this paper we proposed a DSM-based framework to represent specialization relationships that occur in conceptual design. We also proposed an approach to represent both types of relationships, namely, decomposition and specialization in the united framework. The fundamental difference of specialization from decomposition has been explained. We demonstrated how the information contained in the united framework could be effectively used to estimate the formal conceptual similarity between alternative concepts.

This work might have several forms of utility. One of its useful properties is the ability to systematically narrow down the set of alternative concepts for a given solution-neutral problem. By encoding the specialization and decomposition information about alternative concepts in DSM/DMM-supported matrices the system architect can keep track of concepts development during conceptual design phase. Another utility is that the quantitative measure enables to estimate the conceptual similarity between alternative solutions on a conceptual level.

References

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, 292-306.

Chiriac, N., Hölttä-Otto, K., Lysy, D., and Suh, E.S., 2011. Three approaches to complex system decomposition. In DSM 2011: Proceedings of the 13th International DSM Conference, Cambridge, MA, USA.

Crawley, E., Cameron, B., Selva, D., 2015. System architecture: Strategy and product development for complex systems. Prentice Hall Press.

Crawley, E. and Colson, J., 2007. The projection relationship between object process models (OPM) and design system matrices (DSM). In DSM 2007: Proceedings of the 9th International DSM Conference, Munich, Germany.

Danilovic, M. and Browning, T., 2004. A formal approach for Domain Mapping Matrices (DMM) to complement Design Structure Matrices (DSM). In: Proceedings of the 6th Design Structure Matrix (DSM) International Workshop, Cambridge. Cambridge, UK.

Deubzer, F. and Lindemann, U., 2009. MDM application to interrelate hierarchical layers of abstraction. In DSM 2009: Proceedings of the 11th International DSM Conference, Greenville, SC, USA.

Dori, D., 2002. Object-Process Methodology: A Holistic System Paradigm. Springer.

Eppinger, S.D. and Browning, T.R., 2012. Design structure matrix methods and applications. MIT press, Cambridge.

Lindemann, U., Maurer, M. and Braun, T., 2008. Structural complexity management: an approach for the field of product design. Springer Science & Business Media.

Maurer, M. S., 2007. Structural awareness in complex product design. PhD thesis, Technische Universität München, Munich, Germany.

Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.H., 2007. Engineering design – A Systematic approach (3rd Edition). Springer-Verlag London Limited 2007.

Steward, D.V., 1981. The design structure system: A method for managing the design of complex systems. IEEE transactions on Engineering Management, (3), pp.71-74

Contact: Y. Menshenin, Skolkovo Institute of Science and Technology, Skoltech Space Center, Nobelya str. 3, 121205, Moscow, Russian Federation, +7 985 311 5294, y.menshenin@skoltech.ru

20[TH] INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Design Optimization of Size-Adjustable Parts

Georgy Klushin, Clement Fortin, Zeljko Tekic

Skolkovo Institute of Science and Technology

Abstract: This article is devoted to the design with size-adjustable parts approach which is used for mass customization. The article provides a framework for design optimization of size-adjustable parts which enable to achieve incessant adjustability on one or more dimensions of the product. The proposed framework is based on the analysis of size dependences between parts and the difficulty to assemble them addressed through a DSM approach. In addition, these two domains are moderated by the number of size-adjustable parts and the cost of the solution. The article includes a case study to demonstrate the application of the developed framework and a conclusion with a discussion of limitations and directions for further research.

Keywords: Product design, mass customization, design optimization, product variety, design for variety, size-adjustable parts, Design Structure Matrix (DSM)

1 Introduction

In recent decades we can observe the increased demand for customized products, and because of that, the mass customization approach has evolved significantly (Blees et al., 2010; Fogliatto et al., 2012). The core of mass customization is based on the modular design (Tu et al., 2007) and platform approaches (Simpson, 2004). Design objectives for these approaches are usually aimed to increase variety, shorten the lead time and reduce the cost (Simpson, 2004). Nowadays, studies about methods to increase variety emerged as a separate research direction called “design for variety” (Martin and Ishii, 2002).

Studies on variety concern about different parameters which change is required for customization. For example, the scope of adjustable parameters includes, but not limited to, a technology variety (Luh et al., 2011), an application variety (Krause and Eilmus, 2011), as well as size or dimensional variety (Kang and Hong, 2009). The most common approach to achieve variety for these parameters is to design the discrete predefined range of products to fulfill the demand by different configurations (Pahl et al., 2007) or to develop modular building blocks (like Lego bricks), which could be built up during the assembly process to create different variants (Martin and Ishii, 2002). In exceptional cases, adjustability can be achieved by inclusion of the additional functionality in the product architecture to establish adjustable parameter within limits, for example, the seat post clamp of a bicycle (Garneau et al., 2014). However, the dimensional variety represents significant challenges once the range for adjustability becomes uncertain or multi-dimensional variety is necessary. For example, the diversity of different layouts of premises tends manufactures of kitchen units to use parts produced by special order and have dimensions which can be incessantly changed (Figure 1). We refer to this approach as to the design with size-adjustable parts. Existing studies do not cover approaches related to incessant (multi)-dimensional variety and developers usually implement size-adjustable parts on the trial and error basis. The purpose of this article is to provide the framework for implementation of size-adjustable parts in a systematic way for mass customization, and thus, to introduce the new opportunity for developers to make better designs.

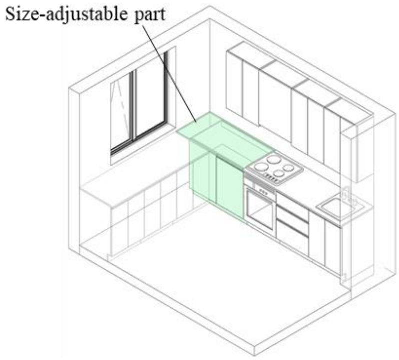


Figure 1. Example of the design with size-adjustable parts (adapted from www.archdaily.com)

To achieve this goal, we should make some deviation from traditional research practice in the design for variety field. Design for variety is mainly considered on a macro level, and the subject of study is a product architecture or a platform in general (Luh et al., 2011; Schmidt III et al., 2008; Suh et al., 2007). Results of these studies provide methods to determine the elements of the platform which should be designed in a way to achieve necessary variety (Li and Azarm, 2002; Simpson et al., 2001; Suh et al., 2007). In comparison to that, studies on a module level (micro level) are rare, for example (Eigner and Zagel, 2007). To have the possibility to consider the use of size-adjustable parts in more details we agreed to focus on a micro level.

Also, we should take into account the difference between two types of variety: spatial (variety in current product) and generational (variety across generations, also known as a temporal variety) (Hsiao and Liu, 2005; Martin and Ishii, 2002). Many industries faced with the shortening of the product life-cycle (Fixson, 2005), for this reason, the generational variety seems to be more promising regarding the lead time and the cost reduction; thus, the dominance of research for generational variety is observed. However, the design with size-adjustable parts should be considered for the current generation of the product, and our contribution would be the rare example of a spatial variety investigation.

To sum up, this article is focused on the incessant dimensional spatial variety on a micro level, or in other words, it is aimed to provide the framework for design optimization of size-adjustable parts. However, the selection of narrow research niche does not prevent that the concepts can also be adapted to the generational variety on macro-level considerations. The rest of the article is organized in the following way, in section two the framework to optimize the design of size-adjustable parts is presented. Section three shows the case-study to demonstrate the application of the developed framework. Finally, in the conclusion section, we explain the results, limitations and future research directions.

2 Framework

This section is devoted to the establishment of the framework for the design optimization of size-adjustable parts. First, we will introduce the theoretical background in the form of causalities between selected design parameters. After that, we will use these connections to describe the optimization procedure for size-adjustable parts, therefore describe the framework.

2.1 Fundamental connections

The fundamental connections appear from the causalities between the number of size-adjustable parts, the number of size dependences per part, the difficulty to assemble for the individual part and the cost of the solution. In this contribution, we focus on the design optimization aimed at reducing difficulty to assemble by splitting size-adjustable parts to decrease the number of size dependencies, the second approach devoted to merging size-adjustable parts retaining the same difficulty to assemble still requires additional investigation and will be addressed in further research.

It is reliably investigated that the cost of the modular product or the platform increases as the number of different parts increases (Hernandez et al., 2003; Pahl et al., 2007; Tu et al., 2007). This artifact can be additionally explained for the case of size-adjustable parts; as we merge two size-adjustable parts on the one dimension we refuse from one machining operation, and in the opposite, the splitting in two parts require additional machining which causes an extra cost. Thus, the first fundamental connection can be stated as: the growing number of size-adjustable parts increases the cost of the solution due to the manufacturing expenses (Figure 2, link 1, “+” (means increase) and “-” (means decrease) notations are adapted from (Blessing and Chakrabarti, 2009)).

Merging (decreasing the number of parts) and splitting (increasing the number of parts) of size-adjustable parts can involve components with different size dependencies (for example, one part has size-adjustability aligned with the width only and second – with the depth only). Thus, the merged part can have greater or equal number of size dependencies, and the split parts can have a reduced or same number of size dependencies (Rajan et al., 2003). In that case, the splitting has a potential to reduce the number of size dependencies per part. Following this explanation for merging and splitting of size-adjustable parts we can formulate the second fundamental connection: by increasing the number of size-adjustable parts (splitting) it is possible to reduce the number of size dependencies per part (Figure 2, link 2).

The growing number of size dependencies for the part may cause the demand for different fastening methods, increase the number of various insertion directions and make assembly path more complex, as a result, the difficulty to assemble this part increases (Boothroyd and Alting, 1992; Sturges and Kilani, 1992). According to this, the third fundamental connection can be expressed as: the growing number of size dependencies per part increases its difficulty to assemble, or, by the reduction in the number of size dependencies for the part it is possible to reduce its difficulty to assemble (Figure 2, link 3).

Finally, the high difficulty to assemble may cause additional expenses for tools and increase the assembly time, thereby, that will enlarge the cost of the solution (Kuo et al., 2001). This observation leads to the fourth fundamental connection: the reduction of the difficulty to assemble may result in cost savings (Figure 2, link 4).

As a result, we have introduced four fundamental connections for design with size-adjustable parts (Figure 2). As we can observe, there is no obvious way to set the optimization path, for example, the growing number of size-adjustable parts, on the one hand, increases the cost, on the other hand, decreases the cost through a reduction of the difficulty to assemble by decreasing the number of size dependencies. To mitigate this contradiction, we will consider the optimization task in a way to find the suitable tradeoff between difficulty to assemble, the number of size dependencies per part and number of size-adjustable parts while assessing the optimized solutions through the cost (Figure 2, dash lines). In such arrangement, the optimization procedure is built around links 2 and 3 (Figure 2) and driven by splitting the size-adjustable parts into several. The results of the optimization are assessed by the cost of the solution (the transition to this stage is made through links 1 and 4) and, if the result of the assessment is successful, the next iteration of the optimization is possible – another size-adjustable part can be split to reduce the difficulty to assemble.

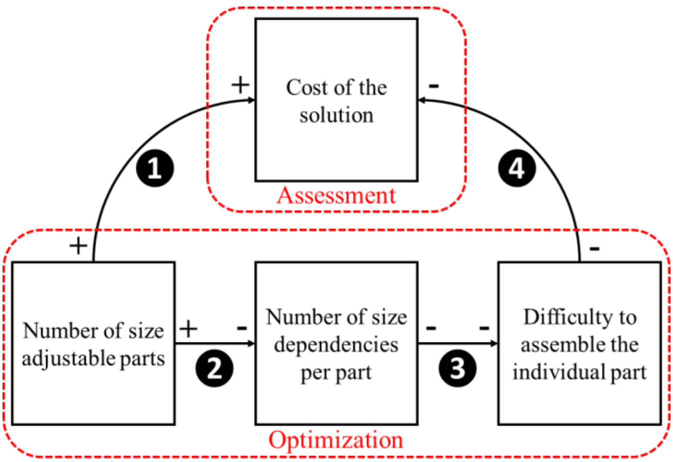


Figure 2. Fundamental connections for design with size-adjustable parts

2.2 Optimization procedure

To arrange a systematic approach for splitting size-adjustable parts to reduce the difficulty to assemble we established the optimization procedure as an iterative process (Figure 3). One iteration of the optimization procedure is devoted to a redesigning one size-adjustable part to reduce the difficulty to assemble by decreasing the number of size dependences for the selected part. The cost is evaluated after each cycle and optimization finishes as size-adjustable parts do not cause difficulties for the assembly process.

The starting point for the optimization procedure depends on the current development phase. If the case is devoted to the design from scratch, then we suggest designing the first version with a minimum number of size-adjustable parts, as the optimization is driven by splitting; else, the current design version can be used as a subject for the optimization.

Cost requirements should be checked before looking for any changes in size-adjustable parts as the optimization driven by splitting can cause additional manufacturing costs. If the solution is out of the budget, then the cost minimization should be performed upfront the optimization (Asiedu and Gu, 1998; Duray et al., 2000; Pahl et al., 2007; Tseng, 1996).

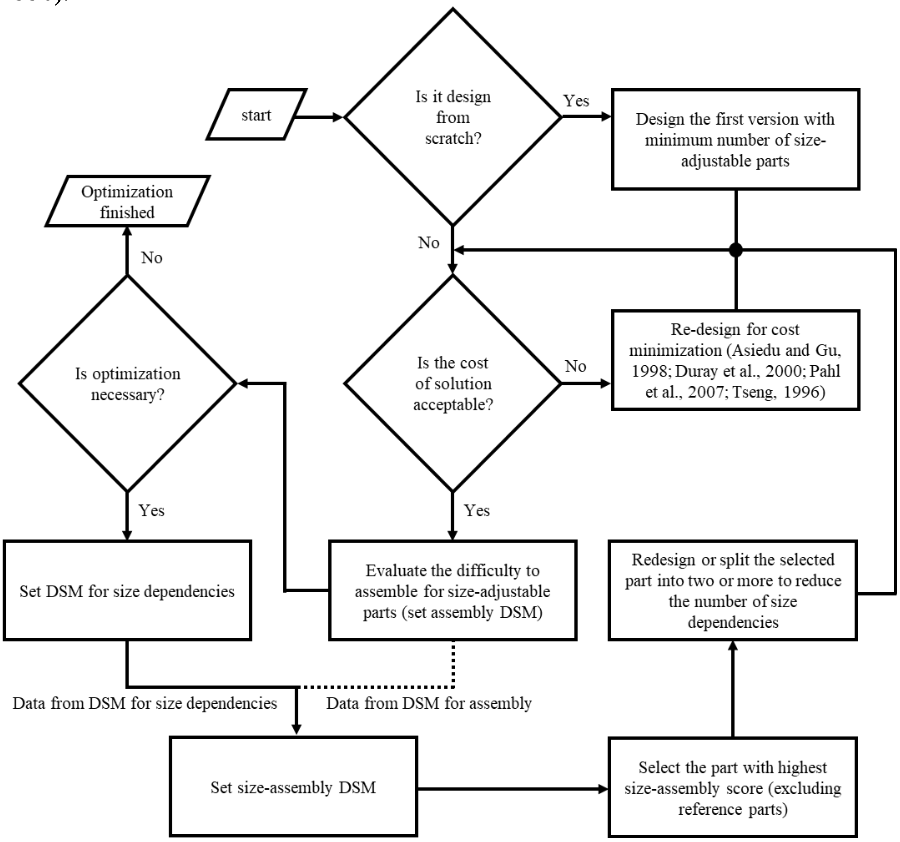


Figure 3. Optimization procedure for size-adjustable parts

The evaluation of the difficulty to assemble serves as a trigger for the optimization. To assess the difficulty, we propose to use the Dependency and Structure Modelling (DSM, also known as the Design Structure Matrix) (Eppinger and Tyson, 2012). Parts in the order of the assembly flow are allocated as headers for columns and rows of the matrix, and then intersection represents the difficulty to assemble between different parts. The four-grade scoring system is adapted from the study of Gunnar Erixon (Erixon, 1996). Each connection is weighted in a color scale, red (= high difficulty to assemble, the assembly path is complex, an additional machining is necessary or tools out of the specification are used), yellow (= moderate difficulty to assemble, tools are required according to the specification), green (= minor difficulty to assemble, no tools are required, or parts are attached indirectly) and blank if parts do not have a connection. The assembly score for each part can be calculated as:

$$Assembly\ score = 9 \times (\#\ red) + 3 \times (\#\ yellow) + 1 \times (\#\ green), \qquad (1)$$

applied to the row of the part ((# color) is the count of color markings in the row). In addition, the average difficulty to assemble is introduced as a mean of assembly scores and can be used to compare design versions. The example of the DSM for the difficulty to assemble can be observed in the case study section (Figure 4a). The markings should be placed based on CAD model investigations, sub-assemblies and prototypes, and several different sizes of the solution should be checked. The red color signals for the necessity of the optimization, however, the difficulty to assemble cannot be used to set optimization path as it is a subjective rating made by few people involved in a design process.

To avoid subjectivity issue and set the optimization path, the DSM for size dependences between parts is introduced (Figure 4b). To acquire data for size dependences, the reference parts should be selected. These parts have only one adjustable dimension. The size dependencies are evaluated in relation to the reference parts and to each other by the following scale, 3 (= parts share the same dimension and geometrically attached), 1 (= the dimension of the part can be calculated from the other, however, parts do not have a direct connection) and 0 if parts are size independent, this procedure gives the objective report about design.

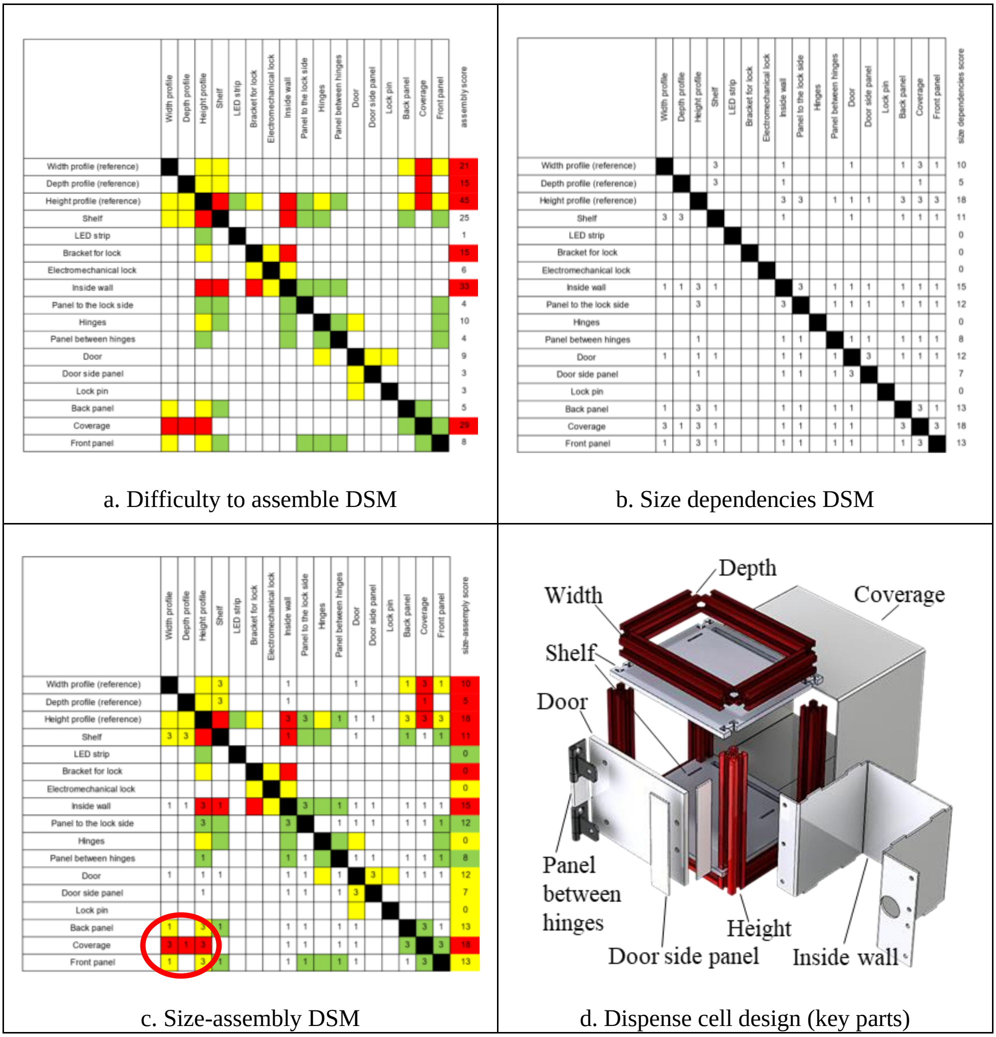
To have a comprehensive picture and select the part for optimization iteration, the two-dimensional size-assembly DSM is introduced (Figure 4c). One dimension is devoted to the assembly difficulty and represented in the color scale, and the second dimension is obtained from the DSM for size dependences between parts. As the size-assembly matrix is set, the size-assembly scores can be calculated as the sum of markings for the individual part and highlighted with the color referred to the highest difficulty to assemble met (red, yellow, then green) (Figure 4c). The part for optimization is selected based on the highest size-assembly score, color then sum of points.

The selected part should be redesigned or split in two or more to reduce the number of size dependences, and the markings in the size-assembly DSM are used to determine design alternatives. After the alternative design was found, the next cycle of the optimization can be started with the cost requirement checking and setting new size-assembly DSM. Once all cases of the high difficulty to assemble for size-adjustable parts are iteratively resolved, the optimization procedure is finished.

3 Case study

The case study is devoted to the dispense cell module of a vending machine (Figure 4d, 5d). Initial marketing investigation revealed 560 different goods which can be potentially sold by the device. Analysis of the package dimensions for these goods provided requirements for inner dimensions of the dispense cell, and the attempt to satisfy these requirements with the multiple size ranges approach would involve the designing of at least 180 size versions. The building block and functional adjustability approaches could not be applied due to security challenges. For these reasons, the design with size-adjustable parts approach was selected, and the goal is to design the dispense cell module that can be produced by special order according to the dimensions of the good to store in this cell.

The initial configuration included 12 types of size-adjustable parts (Figure 4d, three reference parts (highlighted with red color) and nine size-adjustable parts (highlighted with white, front and back panels are not presented)). The prototype revealed several cases where the difficulty to assemble is high (Figure 4a); however, the cost requirements were fulfilled, and thus, the optimization was necessary and possible. The DSM for size dependence was filled (Figure 4b), and it was supplemented by data from the assembly test to set the size-assembly DSM (Figure 4c).



a. Difficulty to assemble DSM

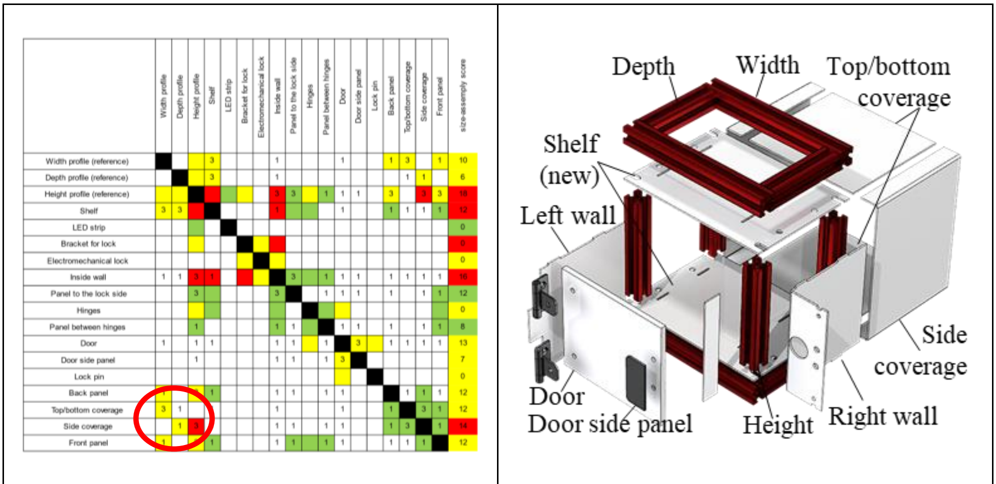
b. Size dependencies DSM

c. Size-assembly DSM

d. Dispense cell design (key parts)

Figure 4. Analysis of the design before the optimization procedure

The “coverage” had the highest size-assembly score =18 and was selected for the first iteration of the optimization (Figure 3). Looking at the row for “coverage” (Figure 4c, circled) we can observe that it has size dependencies with all reference parts, however, the dependency aligned with depth has score “1” compare to “3” for the rest references. Based on this, we can propose splitting into two parts: a first should have dependencies with “width + depth” and a second - “depth + height.” Several design alternatives aligned with this suggestion were generated, and the possible one was implemented. As a result, the “coverage” was split into 2 types of size-adjustable parts (Figure 5c, 5d): the “top/bottom coverage” (determined by width and depth) and the “side coverage” (determined by depth and height). After the “coverage” redesign, the size-assembly DSM for new design version was set (Figure 4c). According to it the “top/bottom coverage” has size-assembly score =12, the “side coverage” =14, compare to 18 for the original “coverage” design.



c. Size-assembly DSM | d. Dispense cell design (key parts)

Figure 5. Analysis of the design within the optimization procedure, c. Size-assembly DSM after the first iteration of the optimization procedure (“coverage” redesign), d. The design of the dispense cell after all iterations of the optimization procedure.

The size-assembly DSM after the first iteration (Figure 4c) is used to find the next candidate for optimization – the “inside wall.” It had the highest size-assembly score for current optimization iteration. Following the same logic as for the “coverage,” the “inside wall” was divided into the left and right parts; thus, the dependency aligned with the width was neglected (compare Figure 4d, 5d). Next iterations of the optimization procedure (Figure 3) were devoted to the “side coverage” (the size dependency with “height profile” was reduced by changing of the connection between profiles), the shelf (the shape was changed to reduce size dependency with “depth profile”). After these four iterations, all cases with the high difficulty to assemble were resolved.

The design optimization of size-adjustable parts made with the help of the optimization procedure in Figure 3 allowed to introduce the design which has 14 types of size-adjustable parts (Figure 5d) and the cost of the solution increased within 1% depending on the size. However, the most important is the fact that the average difficulty to assemble (average assembly score) was reduced almost twice, from 14 (Figure 5a) to 8, and this allowed to shorten the assembly time and minimize the number of defective products.

The case-study also reveals two promising applications for the DSM driven optimization of size-adjustable parts. The first discovery is devoted to the possibility to determine size-adjustable parts which should be redesigned as “one size fits all” or as a set of multiple sizes. The “door side panel” represents the example of such case. According to size-assembly DSM (Figure 4c, 5c), only one cell for the “door side panel” part – the intersection with the “door” part, is devoted to the assembly difficulty and simultaneously represents the highest marking for size dependencies. Based on this observation, we can propose that the parts with only a few markings, which comprise the assembly difficulty and size dependency, in a row of size-assembly DSM are the first candidates to become fixed parts. The second idea relates to merging several size-adjustable parts into one. In comparison to splitting, the merge operation also requires information about materials and manufacturing methods. With this information, it is possible to cluster the DSM to search for merging opportunity. The case study had an intuitive example were the merge was necessary, however, this decision can be also explained with the help of size-assembly DSM. Once the “inside wall” was split into two parts, and a new size-assembly DSM was set, it was observed that the markings in the row for the “left wall” include all markings for the “panel between hinges.” It means that the merging of these parts will not increase the difficulty to assemble, but it will reduce the number of parts, and thus, the cost. These two propositions require in-depth, detailed research to become a part of the optimization procedure for design with size-adjustable parts.

4 Conclusion

This article introduces the design with size-adjustable parts as a solution for the incessant dimensional spatial variety on module level. Analysis of the connections between four objectives of the design with size-adjustable parts revealed the optimization path contradiction (Figure 2). To resolve it, the optimization framework which comprises the optimization procedure (Figure 3) has been developed. The case study of the dispense cell module (Figure 4d) has demonstrated the practical implications of the developed optimization approach. The proposed approach allows systematic implementation of size-adjustable parts for mass customization and introduces the new opportunity for developers to make better designs.

The optimization procedure can be adapted for a generational variety and macro level considerations, however, that may require additional investigation for the size dependencies assessment. The current scale for size-assembly DSM does not distinguish the full scope of cases but serves as a good indicator of the optimization order on module level and supports the search for design alternatives. Transition to macro level or generational variety considerations will require more precise calculations as the number of parts will grow significantly.

Moreover, the current optimization procedure is built upon the splitting of size-adjustable parts to reduce the number of size dependencies, thereby reducing the difficulty to assemble. The merge operation for size-adjustable parts has a potential to reduce the cost of the solution by manufacturing savings and should be investigated in more details. In addition, some of the size-adjustable parts can be redesigned as “one size fits all” and DSM based approach has a potential to determine such parts. These two ideas form a promising research avenue with an aim to establish comprehensive DSM based design optimization for size-adjustable parts.

Finally, during this research, we found that the evaluation of the difficulty to assemble has not been investigated systematically. Individual approaches are available in the literature, and systematic clarification of them will make a significant contribution.

References

Asiedu, Y., Gu, P., 1998. Product life cycle cost analysis: State of the art review. International Journal of Production Research 36, 883–908.

Blees, C., Jonas, H., Krause, D., 2010. Development of modular product families, in: Proceedings of the 12th DSM Conference. pp. 169–182.

Blessing, L., Chakrabarti, A., 2009. DRM, a Design Research Methodology, Springer.

Boothroyd, G., Alting, L., 1992. Design for Assembly and Disassembly. CIRP Annals - Manufacturing Technology 41, 625–636.

Duray, R., Ward, P.T., Milligan, G.W., Berry, W.L., 2000. Approaches to mass customization: Configurations and empirical validation. Journal of Operations Management 18, 605–625.

Eigner, M., Zagel, M., 2007. Product Structures Designed for Variants, in: Proceedings of the 9th International DSM Conference. pp. 249–260.

Eppinger, D.S., Tyson, R.B., 2012. Design structure matrix methods and applications. MIT Press.

Erixon, G., 1996. Modular function deployment (MFD), support for good product structure creation. Proceedings of the 2nd WDK Workshop on Product Structuring, Delft, Holland.

Fixson, S.K., 2005. Product architecture assessment: A tool to link product, process, and supply chain design decisions. Journal of Operations Management 23, 345–369.

Fogliatto, F.S., Da Silveira, G.J.C., Borenstein, D., 2012. The mass customization decade: An updated review of the literature. International Journal of Production Economics 138, 14–25.

Garneau, C., Nadadur, G., Parkinson, M., 2014. Considering human variability when implementing prod. platforms. Advances in Prod. Family and Prod. Platform Design 559–585.

Hernandez, G., Allen, J.K., Mistree, F., 2003. Platform design for customizable products as a problem of access in a geometric space. Engineering Optimization 35, 229–254.

Hsiao, S.W., Liu, E., 2005. A structural component-based approach for designing product family. Computers in Industry 56, 13–28.

Kang, C.M., Hong, Y.S., 2009. A framework for designing balanced product platforms by estimating the versatility of components. International Journal of Production Research 47, 5271–5295.

Krause, D., Eilmus, S., 2011. A methodical approach for developing modular product families. 18th International Conference on Engineering Design, ICED 11, Vol. 44, 299–309.

Kuo, T.C., Huang, S.H., Zhang, H.C., 2001. Design for manufacture and design for “X”: Concepts, applications, and perspectives. Computers and Industrial Engineering 41, 241–260.

Li, H., Azarm, S., 2002. An Approach for Product Line Design Selection Under Uncertainty and Competition. Journal of Mechanical Design 124, 385.

Luh, D.B., Ko, Y.T., Ma, C.H., 2011. A structural matrix-based modelling for designing product variety. Journal of Engineering Design 22, 1–29.

Martin, M. V., Ishii, K., 2002. Design for variety: developing standardized and modularized product platform architectures. Research in Engineering Design 13, 213–235.

Pahl, G., Beitz, W., Feldhusan, J., Grote, K.H., 2007. Engineering Design: A Systematic Approach, Third Edit. ed. Springer, Berlin.

Rajan, P., Wie, V., Campbell, M., Otto, K. & Wood, K., 2003. Design for Flexibility - Measures and Guidelines. International Conference on Engineering Design, ICED 03.

Schmidt III, R., Mohyuddin, S., Austin, S., Gibb, A., 2008. Using DSM to redefine buildings for adaptability, in: Proceedings of the 10th International DSM Conference. pp. 413–425.

Simpson, T.W., 2004. Product platform design and customization: Status and promise. Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM 18, 3–20.

Simpson, T.W., Maier, J.R., Mistree, F., 2001. Product platform design: Method and application. Research in Engineering Design - Theory, Applications, and Methodology 13, 2–22.

Sturges, R.H., Kilani, M.I., 1992. Towards an integrated design for an assembly evaluation and reasoning system. Computer-Aided Design 24, 67–79.

Suh, E.S., De Weck, O.L., Chang, D., 2007. Flexible product platforms: Framework and case study. Research in Engineering Design 18, 67–89.

Tseng, M.M., 1996. Design for Mass Customization. CIRP Annals Manufacturing Technology 45, 153–156.

Tu, Y.L., Xie, S.Q., Fung, R.Y.K., 2007. Product Development Cost Estimation in Mass Customization. IEEE Transactions on Engineering Management 54, 29–40.

Contact: G. D. Klushin, Skolkovo institute of Science and Technology, Nobelya Ulitsa 3, 121205, Moscow, Russia, +7(964)551-58-73, Georgy.Klushin@skolkovotech.ru, www.skoltech.ru

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Supporting Workshop-based Tailoring of Product Development Processes by Metric-based Structural Analysis

Christoph Hollauer, Felix Kölsch, Udo Lindemann

Laboratory for Product Development and Lightweight Design, Technical University of Munich

Abstract: Tailoring complex product development processes for project-specific situations is a task currently inadequately supported and often carried out ad-hoc in companies. Existing approaches in software engineering target the automation of the tailoring activity, which is seen as insufficient in interdisciplinary product development. To address this gap we developed an approach using metric-based structural analysis in order to condense and visualize a process models structural information to support workshop-based collaborative tailoring. The approach has been evaluated using a semi-synthetic test case and an expert interview study.

Keywords: Process Tailoring, Structural Analysis, Project Planning

1 Introduction

Processes play a crucial role in today's product development environment (Bender & Gericke, 2016). They are a critical factor to support engineers in managing increasing requirements regarding customer demands, development costs and time-to-market. Although standard models of product development processes (PDP) are considered useful, they do not have a major added value without adapting the process to the specific context of individual product development (PD) projects (Costache & Kalus, 2011). Consequently, process tailoring is increasingly becoming a focus of process management research (Browing & Ramasesh, 2007), and is addressed in a generic manner in various process standards, such as e.g. ISO/IEC 24748-1 (2010). Nevertheless, in practice, process tailoring is often based on ad-hoc decisions without a systematic approach or support (Pedreiera et al., 2007), although it should be executed in a consistent and systematic manner (Martinez-Ruiz et al 2012). Research has strived to provide corresponding support, mainly in the field of software engineering, focusing primarily on the automated generation of tailored project-specific processes (cf. Hurtado-Alegria, 2014; Park, 2006). However, using automation approaches for tailoring PDPs is at the same time considered difficult to inapplicable (Bender & Gericke, 2016), e.g. due to the (structural) complexity of the PDP models as well as the dynamic context of PD.

Therefore, different alternatives should be explored to support systematic PDP tailoring. One possible approach is the implementation of workshop-based tailoring, including stakeholders affected by tailoring decisions, in order to discuss and collaboratively make tailoring decisions. As a basis for collaborative decision making, profound knowledge is required, e.g. regarding the impact and possible consequences of adaptions in complex process networks, e.g. through the removal of activities. The objective of this paper is to present an analysis framework for condensing and visualizing the information contained in tailoring-relevant knowledge via structural complexity metrics, in order to support workshop-based tailoring. The usage of this information is not limited to the support of individual tailoring decisions, but also used to support the preparation of tailoring workshops in general, e.g. by identifying process stakeholders with common tailoring decisions.

The remainder of this paper is structured as follows: Following the research methodology, the most relevant related work is briefly characterized. Subsequently, the developed analysis framework is described and further explained by concrete application scenarios for supporting workshop-based tailoring. Finally, the evaluation of the analysis concept is presented.

2 Research methodology

This work follows the Design Research Methodology (DRM) (Blessing & Chakrabarti, 2009). The research clarification in this paper is mainly based on previous empirical studies. The second stage, the descriptive study I (DS I), includes reviews regarding relevant topics such as process tailoring and structural metrics, as well as a systematic literature review regarding existing analysis approaches for investigating tailoring knowledge (section 3). The prescriptive study (PS) covers the elaboration of the analysis framework for investigating tailoring knowledge using structural metrics and its application for preparing and conducting tailoring workshops (section 4). The evaluation (descriptive study II, DS II) of the presented concept consists of two parts (section 5): The application evaluation focuses on the applicability of the analysis framework, by testing the approach with a semi-synthetic test case based on real data. An initial success evaluation, investigating the added-value of the developed concept, is conducted via an initial interview-study performed with industry experts.

3 Related work and research gap

The paper at hands presents a systematic approach for analyzing tailoring knowledge. Ginsberg and Quinn (1995) describe tailoring generally as “[t]he act of adjusting the definitions and/or particularizing the terms of a general description to derive a description applicable to an alternate (less general) environment […].”. In the context of PDPs and this work, this is understood as the adaptation of a reference process to a project-specific process applied in a project-specific context. The context of a project can be described by context variables and related values which describe particular specifications (e.g. “project task” and “new development”, “adaptation”). Thereby, dependencies between context values and process adaptations can be modeled as process tailoring rules (PTRs), by including the appropriate tailoring operator (e.g. “select” and “delete”) (cf. Martinez-Ruiz et al 2012, Hurtado-Alegria, 2014). Tailoring knowledge can thus be represented in a rule-based manner and visualized as a graph model using nodes and edges to describe and connect the different entities (e.g. context values, tailoring rules, and process elements).

Utilizing this rule-based representation between context and process model, research has focused on creating tools for automating process tailoring. Different techniques (e.g. feature-based tailoring, neural networks, …) have been applied mainly in software development (cf. Kalus (2013), Park (2006)). However, due to dependencies between context values, their dynamic change over time, and the complexity of PD, adapting the PDP using a configurator with predefined tailoring characteristics is considered not possible (Bender and Gericke, 2016), thus requiring alternative approaches to perform tailoring in a more flexible and interactive manner. A possible concept is to discuss process adaptations during specific tailoring workshop. In order to implement such workshops, a sound basis for decision-making has to be provided by analyzing, condensing and visualizing available tailoring knowledge, due to the structural complexity of PDPs.

Based on this insight, a systematic literature review has been conducted to identify existing approaches for analyzing tailoring knowledge. This procedure did not yield sufficient results for further investigation, indicating that so far little research has been done on this topic. In order to verify this conclusion, the systematic literature review has been modified to enlarge its focus, changing the objective to identifying approaches for analyzing rule-based knowledge in general. As tailoring knowledge can be represented in a rule-based manner, the two systematic reviews are still thematically connected. Nevertheless, expanding the focus of investigation did not increase search results. Most of the identified sources addressed analyzing knowledge transfer in social networks. Hereby the objective is to describe the knowledge flow within an organization by analyzing structural characteristics of the network.

The structural characteristics considered in social network analysis (e.g. centrality) are based on the mathematical fundaments of graph theory and can be transferred to other disciplines as well. An approach for investigating a PDP using graph and network theory by computing structural complexity metrics is presented in detail by Kreimeyer (2009). With the aid of test cases, Kreimeyer (2009) shows that it is possible to evaluate the relevance of individual process elements on a quantitative basis by analyzing the structure of a graph-based PDP model. Since the PDP is the main subject of the tailoring process, the approach presented by Kreimeyer (2009) provides an initial starting point for systematically analyzing tailoring knowledge using structural metrics.

To summarize, tailoring a structurally complex PDP to a project-specific context is complex and knowledge-intensive. Existing tailoring approaches relying on automation techniques focus on “producing” a project-specific process, are limited in terms of applicability due to the software required, and do not foster communication between project stakeholders during tailoring. Tailoring PDPs however requires the inclusion of a multitude of relevant project stakeholders in a collaborative manner, e.g. through workshops. Since the PDPs to be tailored can be quite complex, a systematic approach is needed to analyze and prepare the tailoring knowledge, contained e.g. in the PDP model, required in order to provide a sound basis for the decision-making during tailoring. Approaches for the systematic analysis of tailoring knowledge as well as workshop-based collaborative tailoring are currently lacking. A metric-based structural analysis of graph-based modeled tailoring knowledge provides a starting point for such analyses.

4 Design support: Metric-based structural analysis framework

In order to enable workshop-based tailoring, a five-step methodology has been developed, consisting of the following phases: Preparation, information acquisition, modeling tailoring knowledge as a graph-based "tailoring system model" (TSM), analyzing the TSM, and operationalization of the results in tailoring workshops (cf. Hollauer et al 2018). This paper focuses on presenting the structural analysis of the TSM and thereby support the preparation and realization of tailoring workshops. The analysis consists of four consecutive steps (cf. Figure 1) and has been implemented as a demonstrator using the software Soley Studio (www.soley.io).

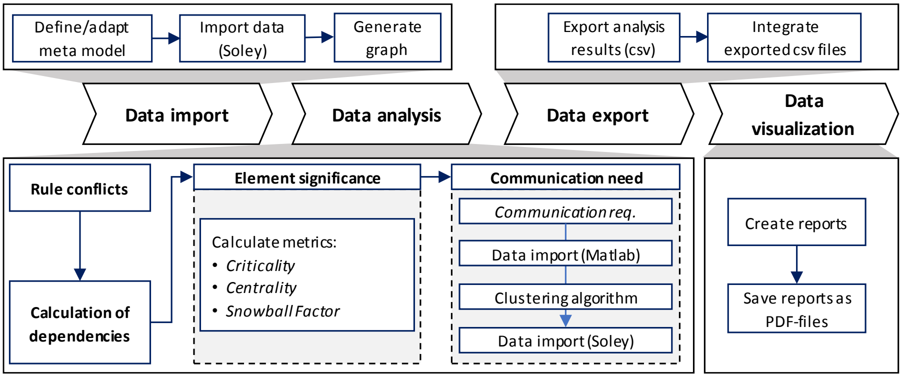


Figure 1. Overview of the analysis procedure

Provided the tailoring knowledge has already been acquired and modeled as the graph-based TSM, the first step of the systematic analysis procedure is to import relevant tailoring knowledge into the analysis tool, modifying the underlying meta model if necessary. The meta models node and edge types are presented in Figure 2.

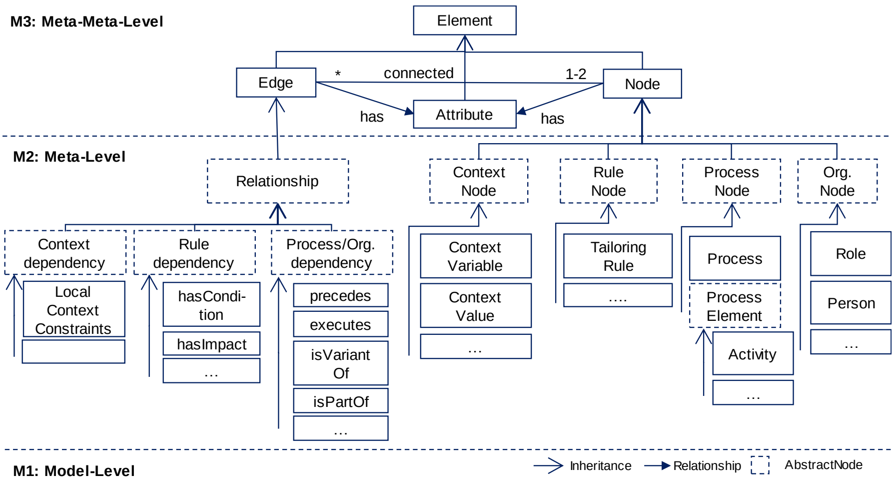
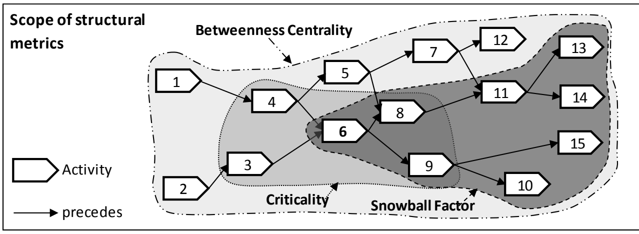


Figure 2. Meta model class diagram for documenting the tailoring knowledge within the four domains Context, Process, Organization, and Rules (excerpt from Hollauer et al 2018)

The model (nodes and edges) can be stored as a csv-file and subsequently imported into the analysis tool which enables visualization in form of a graph and further computational analyses. The actual structural analysis is then carried out on the graph-based TSM using graph rewriting (cf. Helms 2013 XXX). In order to support workshop-based tailoring, the data analysis contains four major parts which are: identification of rule conflicts, calculation of indirect dependencies, calculation of element significance and derivation of communication need among tailoring-afflicted project stakeholders.

PTRs can cause potential *conflicts*. Examples are process elements which are simultaneously impacted by PTRs with different tailoring operators (e.g. “delete” vs. “select”) and a process element variant which is selected by one PTR although an incident and superordinate element is removed by another PTR. Such conflicts can be automatically identified through pattern matching and subsequently, e.g. by adding conditions between context factors which ensure that only one of the corresponding PTRs can be selected simultaneously. Subsequently, *indirect dependencies* between different nodes can be calculated and investigated. On the one hand, indirect dependencies between elements can be used for the metric calculation, on the other hand, the dependencies themselves can be transformed to analytical characteristics of the graph model (e.g. responsible activities per person). Three key *structural complexity metrics are calculated* in order to assess the significance of individual process elements within the PDP, in particular when changes are made to these process elements. These metrics are: *Criticality*, *Snowball Factor* and *Betweenness Centrality* (Figure 3).



Structural metrics of node 6

Criticality (Cri):	4
Snowball Factor (SF)	10
Betweenness Centrality (Cen):	29

Algorithm for calculation per metric	Description	Structural significance	Sources
$Cri = active\ sum \ * passive\ sum$ (1) *active sum: number of outgoing edges* *passive sum: number of incoming edges*	Product of number of incoming and outgoing edges.	Relevance of an element within its immediate environment.	Lindemann et al. (2009)
$SF = \sum_{i=1}^{height} \frac{width(i) * height}{i}$ (2) *height: height of the hierarchy; width: width of the hierarchy; i: hierarchy level*	Sum of product of width per hierachy level and total height, weighted according to inverse of shortest path length to the investigated node.	Relevance of an element for subsequent activities.	Kreimeyer (2009)
$Cen(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$ (3) *s,t,v: nodes of the node set V* $\sigma_{st}(v)$*: no. of shortest paths between s and, including v* σ_{st}*: total no. of shortest paths between s and t*	Sum of number of shortest paths between a pair of nodes including the investigated node weighted according to the total number of shortest paths betweeen this pair of nodes.	Relevance of an element within the entire network.	Brandes (2001); Freeman (1977)

Figure 3. Overview of selected structural metrics quantifying the relevance of process elements with equations 1-3 for metric calculation

The metrics indicate the importance of an individual process element within three different scopes (cf. Figure 3, including formulas 1 to 3 for calculation). This enables the consideration of the process element significance within different neighbourhood sizes during the interpretation of the analysis results, as relying only on a single metric can lead to incorrect conclusions. Calculating the metrics for all elements of the PDP subsequently allows to draw conclusions about the relevance of PTRs. A PTR affecting process elements with high values for *criticality, snowball factor* and *betweenness centrality*, has a potentially large effect on the process. Based on this data, the importance of a PTR can be determined by calculating the mean of each metric for the impacted process elements. Analysing the relevance of single process elements and PTRs, is followed by the fourth stage of the data analysis: *Identifying the need for communication* between project stakeholders regarding tailoring decisions. The need for communication is made up of both process-related and organizational aspects (cf. Heimberger 2017). In our case, process-related communication needs are determined by calculating the number of PTRs affecting two particular stakeholders (via their activities), weighted by the mean metrics per PTR. Therefore, two individuals have a high need for communication, if they have many PTRs in common, which in turn have a large effect on the process. The organizational aspect is based on the fact that the quality of knowledge exchange decreases with increasing (organizational) distance between two stakeholders (Muyun, 2017). The need of communication thus correlates with the distance between two stakeholders within the organizational hierarchy. Combining process and organizational aspects, equation 4 can be formulated to calculate the requirement of communication (RoC).

$$RoC = (\alpha + \beta + \gamma) * (Number\ of\ common\ PTRs) * (Organisational\ Distance)^2 \qquad (4)$$

$$With:\ \alpha = \frac{\emptyset Cri}{max(\emptyset Cri)};\ \beta = \frac{\emptyset SF}{max(\emptyset SF)};\ \gamma = \frac{\emptyset Cen}{max(\emptyset Cen)}$$

Based on the calculated RoCs for each stakeholder pair, a square RoC matrix can be derived and clustered by importing the generated analysis data in a software tool which supports clustering algorithms (e.g. Matlab). After the analysis procedure has been executed, all relevant analytical characteristics of the TSM required for planning and executing of tailoring workshops have been determined. Thus, the data is exported for further processing and visualization.

In order to support the preparation and execution of tailoring workshops, the analysis results are further prepared and visualized (using Excel-based VBA macros in our demonstrator). Consequently, seven types of analysis **reports** with different levels of detail are generated (cf. Figure 4). These reports are grouped into three categories: network level, cluster level, and node level. Reports on **network level** contain information about all nodes of a particular type and give an overview about these elements. Regarding preparing and conducting tailoring workshops it is useful to have such reports for elements of the node class **PTR** and **Stakeholder**. The network-level PTR report contains all PTR nodes including information about the calculated metrics and dependencies between PTRs. Thus, the data sheet enables the identification of outliers and possible errors during modeling on the one hand, and the prioritization of rules based on their effect on the process on the other. In addition, the stakeholder report contains the number of related activities and dependent rules per individual as well as the corresponding cluster assignment. This enables the identification of key stakeholders who need to be involved in the tailoring process and the division of stakeholders into workshop groups (clusters). Reports on **cluster level** then contain information about PTRs which have to be decided during a workshop. Due to the generated metric data regarding the relevance of individual rules, prioritizing the PTRs becomes possible, enabling the derivation of an agenda for each tailoring workshop. To support the decision-making process during such a meeting, reports at **node level** provide detailed information about individual elements (**context**, **PTR**, **process** or **person**). Whereas the reports of the node classes PTR, process element and context mainly serve as reference basis, the stakeholder reports at node level can be used as individual preparation material because they contain all relevant tailoring information (e.g. dependent PTRs, responsible process elements and requirements of communication with other stakeholders) from a particular person's perspective.

However, not every report type is of equal interest to every involved stakeholder, as different stakeholders can assume different **roles** during the tailoring process. Within the scope of this work the three roles **"tailoring expert"**, **"tailoring organizer"** and **"tailoring stakeholder"** are defined. Tailoring experts have a detailed understanding of acquiring and modelling tailoring knowledge and the significance of structural metrics. The reports on network level as well as the node specific reports support the role owner(s) in modelling the tailoring knowledge as well as assisting the workshop participants and moderators during the decision-making process. Tailoring organizers do not require detailed knowledge of graph modeling but must be familiar with the significance of the calculated metrics. Using this knowledge and the stakeholder report on network level, the tailoring organizers can determine appropriate workshop participants. In addition, an agenda for each meeting can be derived with the help of the cluster specific reports. Most of the people involved belong to the "tailoring stakeholder" role (participants of the design process/project) and actively participate in the workshops. This includes the discussion of individual tailoring decisions and submission of a decision recommendation. To prepare for workshops, the tailoring stakeholders can use the stakeholder reports on node level, to familiarize themselves with the relevant PTRs and discussion partners. Thus, the analysis results support the documentation and generation of knowledge, division of workshop groups, development of agendas for meetings and training of involved persons during the preparation of workshops, and decision making regarding process-adaptations during the workshop.

5 Evaluation and discussion

The analysis approach presented in section 4 is evaluated in two ways: First, the functionality of the analysis framework is tested using the developed demonstrator applied on a semi-synthetic test case consisting of real-world PDP data. Missing data (e.g. organizational structure) is generated for the evaluation. After importing the data, the graph-based model consists of 948 nodes and 1553 edges (Figure 4). Performing the four steps of the computational graph enables the automated generation of the reports regarding the different levels of detail (Figure 4). In order to further customize the report generation, an interface allows the selection of specific reports to be generated. The test case confirms the assumption that the analysis framework enables the analysis of complex tailoring knowledge and the condensed visualization with user specific reports.

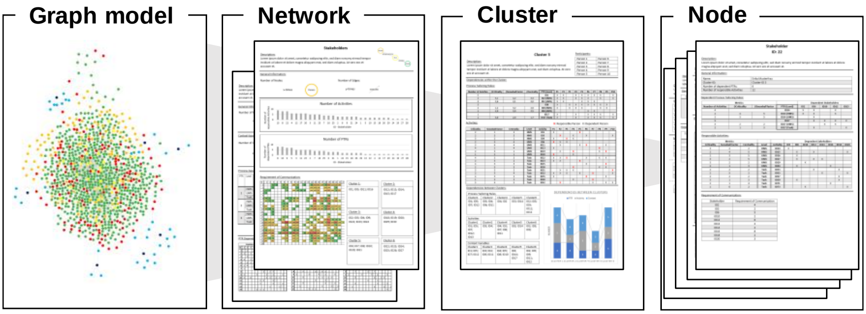


Figure 4. Overview of test case graph model and report types (templates)

Second, the applicability of the approach and its potential added value for workshop-based tailoring has been evaluated via an interview study. The analysis approach and the application of the results (reports) in the context of workshop-based tailoring has been presented to 11 industry professionals. During the semi-structured presentation and interview, discussion with the interview partners produced immediate qualitative feedback. In addition, a questionnaire with 22 question items was handed out after the interview, with eight questionnaires returned. Besides descriptive questions regarding the experts' background, the evaluation form consists on the one hand of questions about the necessity of a systematic support regarding tailoring and on the other hand of an evaluation of the presented approach and analysis results (using five-step Likert scales with 1='strongly agree' and 5='strongly disagree'). Structuring of the tailoring process, internal coordination regarding process adjustments, consideration of dependent stakeholders, complexity of the tailoring process and estimation of the effects of tailoring decisions are all considered challenging by the experts. The quantitative assessment of the added value of the presented reports is shown in Figure 5. In particular, the derivation of suitable workshop groups as well as the metric-based structuring and prioritizing of the rules are to be emphasized positively. The potential of the reports with regard to the other evaluation criteria is also classified as tending to exist. Furthermore, it should be noted that the benefit of the analysis results in the preparation and implementation of tailoring workshops were more appreciated by experts with previous experience in tailoring ($\emptyset = 2.0$; $N = 4$) than by interview partners without experience ($\emptyset = 2.5$; $N = 4$). Besides the quantitative evaluation results, the following points of criticism must be noted from the findings of the qualitative questions and the open discussion:

- The concept assumes that all data is available at a certain level of detail.
- Certain basic knowledge is required to use the reports, requiring additional training.
- The applicability of the concept depends on the size (or duration) of the project. With small projects, the ratio between effort and benefit deteriorates.

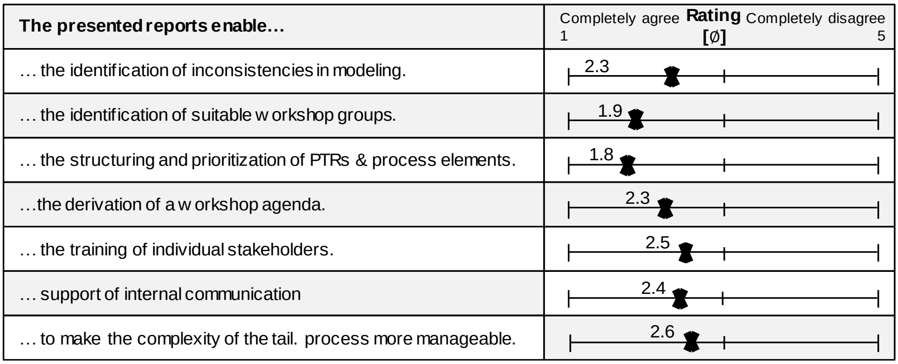


Figure 5. Quantitative evaluation results of the questionnaire regarding the assessment of the presented reports.

The application of the analysis framework requires the availability of an initial data basis, with the data quality being a decisive factor for the quality and value of the analysis results. However, the criticism regarding training can be mitigated, as the training can be adapted to the task of the respective roles. The criticism regarding the relationship between the benefits of the concept and the size of the respective project is countered by automating the analysis.

The presented approach represents a step towards using established structural analysis techniques to support the organization of and decision making during collaborative tailoring of complex PDPs. Using the reports, practitioners can increase transparency regarding tailoring decisions in the complex network structures of PDPs. For example, by ranking PTRs according to impact and identifying communication needs, the tailoring activity can be made more efficient, reducing communication errors, which is not possible using a purely automated approach which solely focuses on the "production" of a project-specific process and does not integrate relevant stakeholders.

6 Summary and future work

This paper presents an analysis framework to quantify the structural characteristics of tailoring decisions and relevant PDP properties using selected structural metrics. This allows to support the design and execution of workshop-based tailoring by identifying communication needs among tailoring stakeholders and providing decision makers with relevant, condensed information regarding the complexity of individual tailoring decisions. Tailoring workshops then allow a collaborative approach for adapting PDPs. A software demonstrator has been implemented and tested, showing the successful automated generation of user-specific reports. In addition, the initial success evaluation indicates that the analysis results create added value for workshop-based tailoring. Nevertheless, points of criticism and limitations exist, which create room for improvement. A first step of future work is the end-to-end application of the design support including the analysis framework in industry. This may require adapting the selected structural metrics and refining the formula for calculating communication requirements. However, more empirical data regarding workshop-based tailoring is necessary in order to test and compare further structural metrics and algorithms. In further steps, a training concept is required, as is a more interactive software support.

References

Bender, B. & Gericke, K. (2016). Entwicklungsprozesse. In U. Lindemann (Hrsg.), Handbuch Produktentwicklung (S. 401-424). München: Hanser.

Blessing, L. T. & Chakrabarti, A. (2009). DRM, a Design Research Methodology. London: Springer London.

Brandes, U. (2001). A faster algorithm for betweenness centrality. Journal of mathematical sociology, 25(2).

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.

Costache, D., Kalus, G. & Kuhrmann, M. (2011). Design and Validation of Feature-based Process Model Tailoring - A Sample Implementation of PDE. Proceedings of the 19th ACM SIGSOFT < and the 13th European conference on Foundations of software engineering, 464-467.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. Sociometry, 35-41.

Ginsberg, M. P. & Quinn, L. H. (1995). Process Tailoring and the Software Capability Maturity Model. Advance online publication. https://doi.org/10.21236/ADA302689

Heimberger, N. (2017). Strukturbasierte Koordinationsplanung in komplexen Entwicklungsprojekten. Dissertation. Technische Universität München.

Hollauer, C., Langner, M. & Lindemann, U. (2018). Supporting Tailoring of Complex Product Development Processes: An Approach Based On Structural Modelling and Analysis. INTERNATIONAL DESIGN CONFERENCE - DESIGN 2018 -

Hurtado Alegría, J. A., Bastarrica, M. C., Quispe, A. & Ochoa, S. F. (2014). MDE- based process tailoring strategy. Journal of Software: Evolution and Process, 26(4), 386-403.

ISO/IEC TR 24748-1, 2010. Systems and software engineering -- Life cycle management -- Part 1: Guidelines for life cycle management. International Organization for Standardization.

Kalus, G. (2013). Projektspezifische Anpassung von Vorgehensmodellen: Feature-basiertes Tailoring. Dissertation. Technische Universität München.

Kreimeyer, M. F. (2009). A Structural Measurement System for Engineering Design Processes. Dissertation. Technische Universität München.

Lindemann, U., Maurer, M. & Braun, T. (2009). Structural Complexity Management: An Approach for the Field of Product Design. Berlin, Heidelberg: Springer Berlin Heidelberg.

Martinez-Ruiz, T., Munch, J., Garcia, F. & Piattini, M. (2012). Requirements and constructors for tailoring software processes: a systematic literature review. In: Software Quality Journa (20), pp. 229–260. DOI: 10.1007/s11219-011-9147-6.

Muyun, S. (2017). The Relation between Organizational Network Distance and Knowledge Transfer Based on Social Network Analysis Method. International Journal of Emerging Technologies in Learning (iJET), 12(6), 171-177.

Park, S., Na, H. & Sugumaran, V. (2006). A semi-automated filtering technique for software process tailoring using neural network. Expert Systems with Applications, 30(2), 179-189.

Pedreira, O., Piattini, M., Luaces, M. R. & Brisaboa, N. R. (2007). A Systematic Review of Software Process Tailoring. ACM SIGSOFT Software Engineering Notes, 32(3), 1-6.

Contact: C. Hollauer, Technical University of Munich, Laboratory for Product Development and Lightweight Design, Boltzmannstraße 15, 85748 Garching bei München, Germany, +49 89/28915136, +49 89/28915144, christoph.hollauer@tum.de

Part VI: Poster Session

20[TH] INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Context-oriented Modularization of Product Development Processes using Matrix-Based Clustering

Christoph Hollauer[1], Robin Thomas[1], Donna H. Rhodes[2], Udo Lindemann[1]

[1]Laboratory for Product Development and Lightweight Design, TUM
[2]Systems Engineering Advancement Research Initiative, MIT

Abstract: System modularization is a common and well-established approach to reduce system complexity. However, methodical approaches for the modularization of product development processes (PDPs) can hardly be found in the literature. The work that exists focuses only on interdependencies between process elements when modularizing the process. This paper proposes a modularization method for PDPs, basing the modularization on the context of the respective process, while still also taking process-internal interdependencies into account. The matrix-based approach applies a clustering algorithm that uses process context data to group process elements into modules. The modular PDP can then be tailored into project-specific PDPs based on the project context at hand. The design and application of lean and efficient project-specific PDPs has promising potential to reduce product development effort and costs.

Keywords: Product development process, modularization, clustering, context, modular processes, tailoring

1 Introduction

Technology companies around the world face challenges like rapidly increasing product and service complexity, increasing customer requirements and numbers of stakeholders involved, as well as shorter development- and product life-cycles (Allweyer 2005, Browning and Ramasesh 2007, Junge 2013, Fischer 2015). Therefore, product development processes (PDPs) that provide the desired outcomes for different product development situations and project scopes in a quick and efficient manner gain more and more importance and are a key aspect of success for every company involved in product development (Sered and Reich 2006, Cooper 2014). As PDPs are highly influenced by the boundary conditions of the development situation at hand and the corresponding specific requirements (Roelofsen 2009), it is recommended to always design the process with its application context in mind. Rosemann and Recker defined a company´s context as the combination of all situational circumstances that impact process design and execution (Rosemann and Recker 2006). Considering a company´s context, a modular PDP can be designed (Rosemann and Recker 2006), which can then be tailored into efficient project-specific PDPs by applying guidelines based on the project-specific situation with all its requirements and constraints (Ginsberg and Quinn 1995, Hollauer and Lindemann 2017). Ginsberg and Quinn define tailoring as "the act of adjusting the definition and/or particularizing the terms of a general description to an alternate environment", which for the area of product development can be interpreted as the adaption of a company's standard set of processes to specific project contexts defined by particular context variables (Ginsberg and Quinn 1995, Hollauer and Lindemann 2017).

This paper presents a methodical approach for modularizing an existing PDP based on the project contexts it is to be tailored to, thereby deducing a modular process that is tailorable into project-specific processes. The following section defines the objectives of the development of the modularization approach and presents necessary theoretic groundwork based on an in-depth literature review. Subsequently, the modularization approach itself and its first evaluation via case studies conducted with an implemented software prototype are detailed. In a final step, potential future research regarding the modularization approach is outlined.

2 Background and Objectives

This section provides a brief overview of modularity in general and modularization in the areas of processes and products to then derive the objectives of the context-oriented modularization approach for PDPs.

2.1 Modularity

A broad variety of definitions of modularity can be found in the literature. Reijers, Mendling et al. (2010) propose a very general definition, stating that modularity is commonly interpreted as the design principle of having a complex system composed from smaller subsystems, that can be managed independently yet function together as a whole. Göpfert (1998) and Bauer (2016) describe modularity as an approach to reduce the complexity of a system by dividing it into smaller subsystems or modules, that minimize interfaces between each other, but have a high degree of interaction within each module. The concept was first used for product modularization in order to be able to understand and control the steadily growing complexity of products, preceding its application on processes (Göpfert 1998, Langlois 2002). Besides the reduction of system complexity as the general motivation for modularity, further advantages are: standardization, decoupling, combinability, flexibility, reuse, efficiency, controllability, replaceability, changeability and adaptability (Sanchez and Mahoney 1996, Gu et al. 1997, Göpfert 1998, Gu and Sosale 1999, Renner 2007, Seol et al. 2007, Krause and Ripperda 2013).

2.2 Existing approaches

A literature review regarding basic information and existing approaches for process modularization was conducted with regard to PDPs (focus area) and business processes (BP). BP are usually less complex, less parallel, include less iterations and have less complex interdependencies within the process (Browning et al. 2006, Lindemann 2009, Clarkson and Eckert 2010, Koch 2015), making modularization easier and more common.

PDP modularization: The investigation of modularity in the area of PDPs identified a number of methods and approaches for flexible design of PDPs due to the respective development situation. Examples are a method of modelling PDPs using process blocks (Bichlmaier et al. 1999), relation-oriented process synthesis (Baumberger 2007), the FORFLOW process model (Roelofsen 2011), and the Stage-Gate approach (Cooper

2001). All of these and other existing methods in the literature are well-established tools providing useful general information and guidelines for the design of flexible PDPs based on different development situations. However, no methodology for modularizing an existing process based on a company's different project contexts could be found. The only method coming close is the concept for design process modularization proposed by Seol et al. (2007). The authors divide an existing PDP into its constituent activities and cluster (group) these into modules via an algorithm analyzing the process flow between the activities in a design structure matrix (DSM) (Seol et al. 2007). The modularization approach presented in this paper similarly divides the overall process into activities and groups them into modules with a clustering algorithm, but the method proposed by Seol et al. (2007) could not be used as the basis for the development of a context-oriented modularization method. The reason for this is that the algorithm they apply is too limited and the modularization is purely based on the process flow, not taking project contexts into account. Nevertheless, ideas, requirements and restrictions could be derived from that concept. As there is no methodical approach for a context-oriented modularization of existing PDPs, a second step was to analyze modularity in the area of business processes, where its application is more established.

Business process modularization: The main purposes of the modularization of business process are to increase process understanding among the stakeholders (Gruhn and Laue 2006, Mendling et al. 2010), to support communication (Reijers and Mendling 2008, Melissen 2013), and to take advantage of reuse of already existing modules (Gruhn and Laue 2006, Reijers and Mendling 2008). However, in general, the focus of research on business process modularization is of conceptual nature and there are no objective and explicit guidelines, tool support or methodical approaches, that modelers in practice can rely on (Reijers and Mendling 2008, Mendling et al. 2010). The idea of basing process design on the context of a company is outlined by Rosemann and Recker (2006), who suggest designing flexible, context-oriented business processes, but do not propose any kind of methodical approach. To summarize, modularization approaches for business processes that could be applied on PDPs considering the company context are also currently lacking.

Product modularization: With limited existing modularization methods for PDP or business processes to base on, the decision was made to investigate methods for product modularization for adaptation to the use on PDPs. This was chosen as a significant number of elaborated modularization methods for products are readily available, and the application of product modularization is a very common approach (Krause and Ripperda 2013). After an in-depth review regarding existing product modularization methods and a detailed comparison of the eight most promising approaches, the extended modular function deployment proposed by Stake (Stake 2000) was found most promising to be adapted and extended into a context-oriented modularization approach for PDPs.

2.3 Research gap and objectives

Frameworks and guidelines for the design of flexible PDPs already exist, but methodical support for the modularization of an existing PDP based on the process context is currently limited. A modular process could subsequently be tailored into project-specific processes more easily. This paper aims to contribute to closing this research gap by elaborating a corresponding modularization approach, focusing on the following objectives:

- The actual modularization of the PDP should be completely based on the respective company's context data as the key novel aspect of the method.
- The method should take interfaces between process activities into account, to allow a comparison of the quality of different modularization scenarios.
- The method should not be limited to a specific industry sector and specific type or complexity of PDP, in order to maximize the applicability of the method.
- The method should be implemented in a software demonstrator as a proof of concept.
- The software demonstrator should be applied and evaluated using case studies.

3 Proposed method

Figure 1 displays the steps of the final context-oriented modularization method for PDPs and its modularization algorithm after several steps of adapting, extending and modifying the modular extended modular function deployment that served as the basic framework. The individual steps are subsequently detailed.

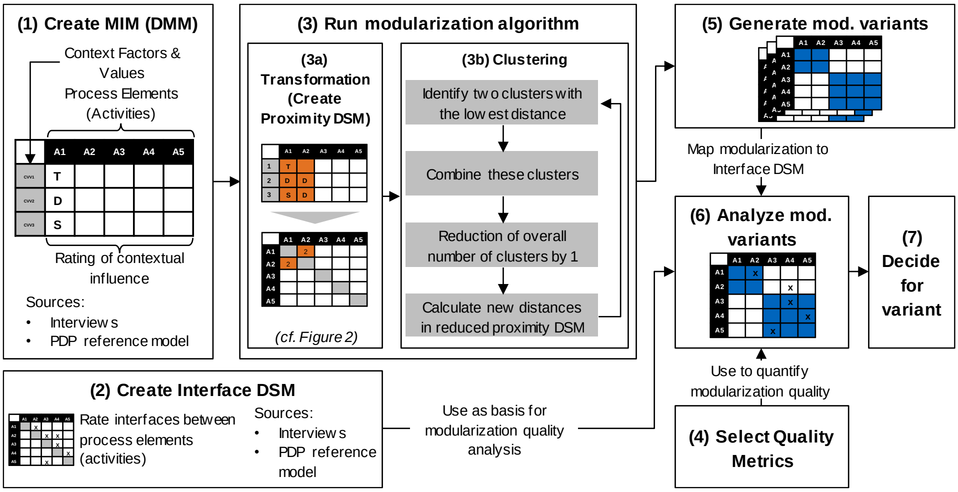


Figure 1: Steps of the modularization method and the modularization algorithm

(1) Create MIM (Rate activities regarding contextual influence): The modularization is based on the company's project contexts, which can be defined as all internal and external boundary conditions influencing the development activities within the respective company. The context is documented in the form of context variables with different values to describe the possible project contexts of a specific company. Examples include the different types of projects that are conducted within a company, the disciplines involved, the industries and markets it is doing business in, the complexity of its product portfolio, etc. The PDP is documented in a reference process model containing, among other, the process activities Subsequently, all process elements (activities) are rated regarding the influence of each context variable value on them in the so-called module indication matrix (MIM). The MIM is a domain mapping matrix (DMM), containing the process elements (columns) from the reference PDP model and the context variable values (rows), acquired using e.g. interviews. The rating regarding the influence of each context value on each process element can be performed via a numerical rating system (strength) or using qualitative operators that are later applied for the tailoring of project-specific processes. For example, for a particular context value an activity “must be tailored”, “is deleted”, or “a specific mode selected” (cf. “T”, “D”, “S” in Figure 1). The rated MIM forms the basis for the modularization algorithm, by comparing the similarity of the ratings of contextual influence on process elements, grouping elements into modules that have similar context ratings and will therefore be necessary in the same project context.

(2) Create Interface DSM (Rate activity interfaces): To consider dependencies between process elements (activities), their interfaces are documented in a design structure matrix (DSM). Different types of process interfaces can be considered, based on an interface catalog derived from literature, e.g. interfaces regarding collaboration, communication, information, and organization. The interfaces must be defined and rated by organizational process experts. The only requirement regarding the rating system applied to quantify the intensity of process interfaces is that it has to be numerical. The completely rated interface DSM (iDSM) forms the decision basis to assess the quality of the various modularization variants generated by the modularization algorithm. As in-depth process knowledge is required to perform both, rating in the MIM and the iDSM, company-internal process experts should rate their respective process activities regarding context influences and interfaces for the application of the method.

(3) Run modularization algorithm: With the rated MIM, a two-step modularization algorithm is run to generate possible modularization variants. In a first step, the MIM is transformed into a symmetrical proximity matrix (pDSM), with which the actual modularization is performed (see figure 2 for a simplified example). The clustering algorithm applied in the software prototype and case studies is a *hierarchical*, *agglomerative* clustering algorithm, which was selected and designed based on the guidelines for the elaboration of clustering algorithms in Backhaus et al. (2015). The application of other clustering algorithms is possible as well, as long as they use the similarity of the process element´s context ratings as the clustering criteria.

(3a) Transformation algorithm: To run the clustering algorithm, the MIM is first transformed into a pDSM (process elements x process elements), containing the distance of the ratings of the process elements from the MIM. During the transformation, each activity pair is compared and the calculated distance documented in the respective cell. For each pair, non-identical context variable values increase the distance by the value “1” (cf. Figure 2, orange highlights). This basic counter can be augmented through multiplication and addition of the basic counter with a pre-defined weighting system (Figure 2, right). The weighting system can be adapted to the situation at hand to increase or decrease the influence of context variable values on the modularization. Possible elements of the weighting element are: the active sum of context variable values in order to increase the weight of influential values, the probability of occurrence for individual context values, or modified distance counters for safety/quality relevant context factors.

(3b) Clustering algorithm: The selected clustering algorithm is subsequently applied on the resulting pDSM (cf. Figure 1). The algorithm starts with the assumption that each matrix element (process activity) forms its own module. In every step, the two elements/clusters with the smallest distance regarding the influence rating of the different context variable values are grouped together into a cluster and the overall number of clusters is reduced by one. Afterwards, the distances of the newly formed cluster to all other existing clusters are updated, leading to a reduced pDSM, upon which the next algorithm step will be executed on. For this step either the smaller (single linkage, SL) or the higher distance (complete linkage, CL) of the two distances of the clusters being combined can be assessed as the new distance to each other element/cluster, leading to different possible modularizations. Each of the procedures or a combination of both can be favorable under certain circumstances, but further research regarding this aspect is necessary. The steps are repeated until a previously defined number of clusters (i.e. modules) is reached. This way, several possible modularizations can be generated and compared to identify the solution with the highest modularization quality due to quality metrics.

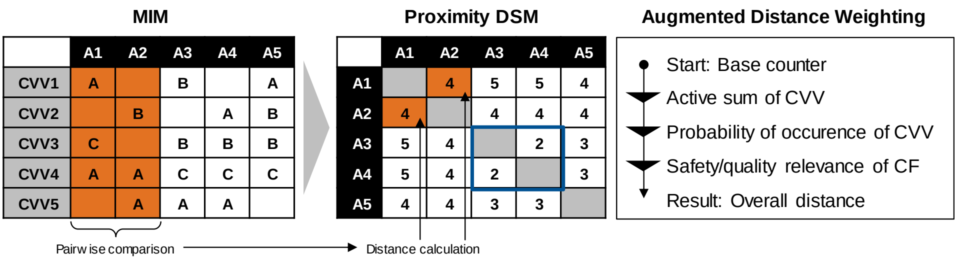


Figure 2: Simplified example of the transformation and clustering algorithm

(4) Iterate to generate modularization variants: In order to identify a high-quality modularization, several iterations of the modularization algorithm are necessary to generate variants for comparison. The preferred modularization solution, showing the highest modularization quality, is subsequently selected using the applied modularization quality metrics.

(5) Select modularization quality metrics: Structural metrics are employed for the comparison of different potential modularizations generated by the algorithm. The metrics are applied on the modularized iDSM. Most modularization quality metrics determine the modularization quality based on an analysis of the interfaces between and within the modules. Additionally, metrics focusing on different modularity aspects, e.g. the number of involved stakeholders per modules, are feasible as well. The metrics should be chosen based on the respective situation and the desired focus. Examples of possible metrics supporting the analysis of the modularization in the iDSM are:

- *Cluster perspective/module density* (Behncke 2017, Koppenhagen 2004): Minimizing unwanted interfaces between modules that can limit the success of modularization.
- *System perspective/module independence index* (Behncke 2017, Koppenhagen 2004): Maximizing necessary interfaces within modules.

- *Module qualities* (Kreimeyer 2009): Analyzing the compactness of modules (interaction of a module with its environment) and the flow of information between them, both of which should be limited for a high-quality modularization.
- *Stakeholder metrics*: Limiting the number of involved stakeholders per module or the number of modules one particular stakeholder is involved in.

One aspect the quality metrics should always consider is the overall heterogeneity of the modules that increases with a decreasing number of modules and increasing number of elements per module.

(6) Analyze modularization variants by calculating quality metrics: The metrics chosen for the analysis of the potential modularizations in a specific context are calculated and compared to provide the data basis for deciding on one of the modularizations. Before applying the quality metrics, each possible modularization generated by the clustering algorithm must be transferred to the iDSM.

(7) Decide on one modularization: Based on the results of the variant analysis using the quality metrics, the last step is making a decision for the design of the modular PDP with the highest quality due to the quality metrics.

4 Evaluation

The modularization method is implemented in a Microsoft Excel-based software prototype programmed using visual basic for applications (VBA). The software prototype was subsequently applied on two case studies to verify the overall approach, including the modularization algorithm as well as the quality metrics. This was done to ensure the algorithm is functioning as intended and provides valid results that comply with the objective of deducting a context-specific modularization of an existing PDP.

The first case study was conducted with a small, academic set of input data with low complexity. In both case studies, modularization variations were automatically generated by the modularization algorithm implemented in the software prototype and manually compared by the authors applying the modularization quality metrics. Figure 3 shows an example of a modularization during the first case study displayed in the MIM. For the academic case study, no expert-based independent evaluation of the results was possible.

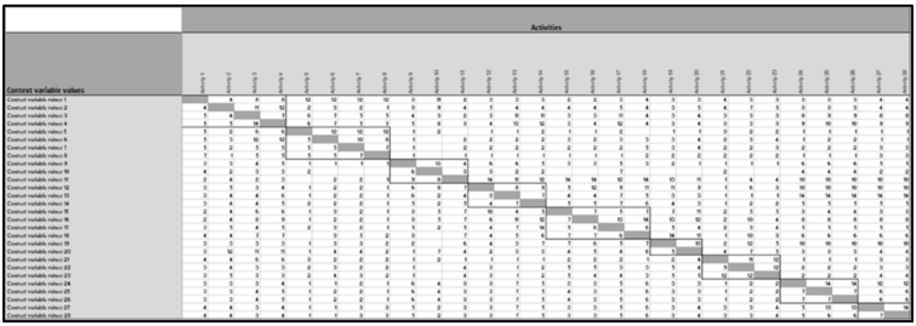


Figure 3: Modularization example for the academic case study

The second case study was based on data from an industrial case study (medium-sized plant engineering company), where 218 process activities and 231 context values with corresponding MIM ratings have been obtained, but only partial data regarding the activity interface ratings (iDSM) was available. The MIM data included ratings in a quantitative form, indicating whether an activity can be dropped, needs to be carried out, or needs to be carried out intensively, depending on the context values for a particular project. Due to confidentiality concerns, this data cannot be published. After performing test runs with both SL and CL algorithms, the heterogeneity curves of the resulting modularization were analyzed, but, due to their similarity, did not provide a conclusive lead for the selection of an algorithm. Also, no optimal number of clusters due to the "elbow-criterion" could be identified (cf. Backhaus et al 2015, pp. 494-496). The eventual clustering of the calculated pDSM was subsequently carried out in two stages to derive 20, 30, 40, and 50 clusters: First, a SL algorithm generated 10 clusters consisting of only one to three elements, with another cluster containing the remaining activities. Removing these cluster, the remaining larger cluster was "sub-modularized" using a CL algorithm. The subsequent metric analysis indicated that the combination of SL and CL algorithms with a cluster count of 20 produced the modularization of the highest quality (not regarding the homogeneity of clusters). However, the choice of algorithm strongly depends on the intended number of clusters, as the two-stage approach only produced the best results for 20 clusters. For higher numbers of clusters, the differences between the combined approach and a single stage CL algorithm were negligible. In fact, if the objective is to derive more homogeneous clusters, for 50 clusters the CL algorithm produced slightly better results, and also requires less effort. The number of intended modules should be defined with the overall number of process activities in mind, setting the number of modules to e. g. 10 to 25% of the overall process activities.

For the second case study, a detailed evaluation with the process expert responsible for the elaboration of the context and process model was performed. The process expert confirmed the usefulness of the modularization metrics and the validity of the results. The most important aspect he pointed out, was the selection of the applied quality metrics. They must be selected carefully regarding the the key objectives of the modularization in a specific situation (e.g. avoiding upstream interfaces possibly causing rework, minimizing the number of stakeholders involved per module, minimizing the overall flow of information between modules, etc.) to assure finding the optimal solution. Therefore, internal process experts should select the quality metrics to apply in the decision-making process, as well as which interfaces to consider for the interface analysis in the iDSM.

To summarize, the case studies showed that the modularization approach provides the necessary tools and guidelines for a context-oriented modularization of an existing PDP and verified the usefulness of the quality metrics for supporting the decision for one of several possible modularizations generated by the modularization algorithms. Additionally, the case studies revealed promising areas of further elaboration of the modularization method. However, the approach is currently considered preliminary and requires further testing and refinement.

5 Conclusion and future research

In this paper we have presented a preliminary approach for an algorithmic, matrix-based modularization of PDPs based on differing project contexts. The approach considers relevant process interfaces by basing the modularization quality assessment on the interface analysis of the modularized PDP. The thus modularized PDP is expected to be more easily tailorable due to the grouping of similarly influenced activities. The process modules serve as a basis for grouping and managing activities subject to similar contextual influences. To summarize, the identified research gap can be addressed by the developed approach, as it reproducibly generates a modular PDP, that capitalizes on the advantages of modularity, such as. adaptability and flexibility. Subsequent tailoring of the reference PDP can avoid unnecessary activities and therefore reduce time, effort and cost. This tailoring step can, for example, be performed in collaborative workshops with project stakeholders. The presented approach can contribute to reducing the process tailoring effort, which is a crucial advantage in times of strong competitiveness in globalized markets and steadily increasing importance of efficiency (Sered and Reich 2006, Fischer 2015). The current state of the developed method represents a basis for further experimentation with a high potential for further elaboration and application in industry.

Additional case studies need to be conducted for further evaluation and refinement, with different input data and boundary conditions. The following aspects should be tested and compared in particular: Different rating systems for the assessment of the influence of the context variable values on process elements (activities) in the MIM, weighting systems for the transformation of the MIM into a DSM, different clustering algorithms, and the significance of the quality metrics for practitioners. The base approach itself is designed to be adaptable to such changes.

Another area for future research is the improvement of the software prototype, both in terms of performance as well as automation of the decision-making process by including the quality metrics in the algorithm to combine the generation of module variations and their analysis (closed-loop optimization). So far, this needs to be done manually, but with an enhanced software tool the user could define quality metrics and weighting system beforehand, with the software automatically generating the modularization solution space, identifying the best solution automatically. Another aspect not yet covered is how to keep the resulting modular PDP up to date and adapt it to significant changes in the context.

References

Allweyer, T. 2005. Geschäftsprozessmanagement: Strategie, Entwurf, Implementierung, Controlling, W3l GmbH.

Backhaus, K., et al. 2015. Multivariate analysemethoden: eine anwendungsorientierte einführung, Springer-Verlag.

Bauer, W. 2016. Planung und Entwicklung änderungsrobuster Plattformarchitekturen, Unpublished Dissertation, Verlag Dr. Hut.

Baumberger, G. C. 2007. Methoden zur kundenspezifischen Produktdefinition bei individualisierten Produkten, Verlag Dr. Hut.

Behncke, F. G. H. 2017. Beschaffungsgerechte Produktentwicklung, Unpublished Dissertation.
Bichlmaier, C., et al. 1999. PMM-Process Module Methodology for Integrated Design and Assembly Planning. Proceedings of the 4th Design for Manufacturing Conference, Las Vegas.
Browning, T. R., et al. 2006. Key concepts in modeling product development processes. Systems Engineering 9(2): 104-128.
Browning, T. R. and R. V. Ramasesh 2007. A survey of activity network- based process models for managing product development projects. Production and operations management 16(2): 217-240.
Clarkson, J. and C. Eckert 2010. Design process improvement: a review of current practice, Springer Science & Business Media.
Cooper, R. G. 2001. Winning at New Products: Accelerating the Process from Idea to Launch.
Cooper, R. G. 2014. What's Next?: After Stage-Gate. Research-Technology Management **57**(1): 20-31.
Fischer, U., Müller, P. 2015. Mega-Trends und ihr Einfluss auf die Produktentwicklung.
Ginsberg, M. P. and L. H. Quinn 1995. Process tailoring and the the software capability maturity model.
Göpfert, J. 1998. Modulare Produktentwicklung. Innovationsforschung und Technologiemanagement, Springer: 139-151.
Gruhn, V. and R. Laue 2006. Complexity metrics for business process models. 9th international conference on business information systems (BIS 2006).
Gu, P., et al. 1997. An integrated modular design methodology for life-cycle engineering. CIRP Annals-Manufacturing Technology 46(1): 71-74.
Gu, P. and S. Sosale 1999. Product modularization for life cycle engineering. Robotics and Computer-Integrated Manufacturing 15(5): 387-401.
Hollauer, C. and U. Lindemann 2017. Design Process Tailoring: A Review and Perspective on the Literature. International Conference on Research into Design, Springer.
Junge, M. 2013. Modularisierung in der Automobilindustrie. Perspektiven und Facetten der Produktionswirtschaft: Schwerpunkte der Mainzer Forschung: 89.
Koch, S. 2015. Einführung in das Management von Geschäftsprozessen: Six Sigma, Kaizen und TQM, Springer-Verlag.
Koppenhagen, F. 2004. Systematische Ableitung modularer Produktarchitekturen. Berichte aus der Konstruktionstechnik, Shaker, Aachen: 16.
Krause, D. and S. Ripperda 2013. An assessment of methodical approaches to support the development of modular product families. 19th International Conference on Engineering Design (ICED13), Seoul.
Kreimeyer, M. F. 2009. A structural measurement system for engineering design processes. Dr. Hut, München.
Langlois, R. N. 2002. Modularity in technology and organization. Journal of economic behavior & organization 49(1): 19-37.
Lindemann, U. 2009. Methodische Entwicklung technischer Produkte. 3., korrigierte Auflage, Berlin, Heidelberg (Springer).
Melissen, R. R. 2013. A business process harmonization approach: using process mining techniques.
Mendling, H. R. J. and R. Dijkman 2010. On the Usefulness of Subprocesses in Business Process Models.
Reijers, H. and J. Mendling 2008. Modularity in Process Models: Review and Effects. Business Process Management: 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008. Proceedings. M. Dumas, M. Reichert and M.-C. Shan. Berlin, Heidelberg, Springer Berlin Heidelberg: 20-35.
Renner, I. 2007. Methodische Unterstützung funktionsorientierter Baukastenentwicklung am Beispiel Automobil.

Roelofsen, J. 2009. Situationsspezifische Prozessplanung und -konfiguration. Flexible Unterstützung in der Produktentwicklung: Prozesse - Daten - Navigation. H. e. a. Meerkamm. Aachen, Shaker: 45-62.
Roelofsen, J. M. K. 2011. Situationsspezifische Planung von Produktentwicklungsprozessen, Verlag Dr. Hut.
Rosemann, M. and J. C. Recker 2006. Context-aware process design: Exploring the extrinsic drivers for process flexibility. The 18th International Conference on Advanced Information Systems Engineering. Proceedings of Workshops and Doctoral Consortium, Namur University Press.
Sanchez, R. and J. T. Mahoney 1996. Modularity, flexibility, and knowledge management in product and organization design. Strategic management journal 17(S2): 63-76.
Seol, H., et al. 2007. Design process modularization: concept and algorithm. Concurrent Engineering 15(2): 175-186.
Sered, Y. and Y. Reich 2006. Standardization and modularization driven by minimizing overall process effort. Computer-Aided Design 38(5): 405-416.
Stake, R. 2000. On conceptual development of modular products-Development of supporting tools for the modularisation process, Industriell produktion.
Yassine, A. A., et al. 2013. Simultaneous optimisation of products, processes, and people in development projects. Journal of Engineering Design 24(4): 272-292.

Contact: C. Hollauer, Technical University of Munich, Laboratory of Product Development and Lightweight Design, Boltzmannstraße 15, 85748 Garching bei München, Germany, +49 89/28915136, +49 89/28915144, christoph.hollauer@tum.de

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

The Algebra, Logic and Topology of System-of-Systems

Mats Johansson[1], Patrik Eklund[1], Jari Kortelainen[2], Michael Winter[3]

[1] Umeå University, Department of Computing Science, Umeå, Sweden
[2] South-Eastern Finland University of Applied Sciences, Mikkeli, Finland
[3] Brock University, Department of Computer Science, St. Catharines, Ontario, Canada

Abstract: In this paper we propose an information structure enrichment of relational models underlying design structure models typically used in System-of-Systems. Such design structures are algebraically, logically and topologically mostly unstructured relations as treated within naïve set theory. The paper also aims to show how an enriched information structure can be applied to monitor the health status of a System-of-System as an alternative to fault trees.

Keywords: Contact lattices, generalized terms, nearness, proximities.

1 Introduction

System-of-Systems embrace several intertwined subsystems and even several interrelated subsystem models involving humans and machines, physics and economics, evaluations and predictions, and many more aspects, all having specific modelling requirements. Systems are designed and manufactured, operated and maintained, and eventually replaced. From a system point of view, and while operational, the lifespan of a subsystem involves condition monitoring, identification of changes, and various aspects and phenomena that needs to be quantified and qualified, often in stochastic and many-valued settings. Monitoring of operations often involves identifying or preventing defect, as a matter of diagnostics. On the other hand, system functioning is important to maintain at required levels, or restored after shutdown or breakdown. Service and maintenance therefore has to focus both diagnostics as well as functioning.

As an example, any system that includes running mechanical components is affected by wear. In many cases, there are predictive models describing the effects of this wear over time and these models are the base for maintenance schedules. In some cases the actual wear of individual components will deviate from the predictive model and in these cases it is useful to have a system that may detect this deviation. Many systems are equipped with different kinds of sensors. In a system-of-systems there may be a number of predictive models that may or may not be similar in kind. There might be models for mechanical wear, models for fluids, air filters etc. Each will contribute to a general representation of the current projected status of a system-of-systems. Many systems are equipped with different kinds of sensors that may be used to detect deviations from the predicted models, to complement the models and to give a better general representation of the system status and not to forget they may be used to make new and better prediction models for maintenance of system and system-of-systems. This means that there is a need to be able to handle sets of data from different models that all aims to express various forms of states, but that are not necessarily using the same terminology. The way in which they differ might be expressed as a distance or rather *nearness*, since they are related in some way. Since a system or a system-of-systems by necessity is contributed by more than one individual part there will most likely be individual aspects that needs to be addressed that may or may not have a high level of nearness from a topological point of view.

To explain this we start with one of the simplest mechanical systems possible. Two gears who's cogs are linking in to each other. They have the exact same amount of cogs, i.e. a 1:1 ratio and are suspended in mid air without any bearings or axels. In this case any observation or prediction model for any of the two gears would be highly identical with the other. Since each cog would touch another cog the same amount of times the wear would be very similar regardless of which gear is chosen for observation or model. The nearness between the sets of terms created would be very close. Anything made to increase the complexity of this simple system will introduce differences between observations made, even if we use the same basic model. Say that we change the ratios between the gears to 2:1. This would mean that each individual cog on one gear will touch a cog on the other gear twice as often during a finite amount of time. This would mean that even if it would be quite possible to use the same basic model to predict the wear of each wheel, we also will have to make and introduce a new model that handles the combined wear of both gears since the increased wear on one gear in fact may affect the other gear as a consequence. In this case it is easy to see the relation, or nearness in the cause and effect between the components regardless of how it is expressed since it is a very small system. If this is scaled up in to a system-of-systems the importance of the use of nearness as a way to express relation is far more important. Since a system-of-systems with high probability will be made using components of different makes and vendors using different kinds of standardizations or even vendor specific notations for diagnostics. There will be a need to express how closely related seemingly different values or terms are. Both to draw conclusions about the current status of a system-of-systems but also to identify unobvious relations that may enable better conclusions about the overall state of a system-of-systems.

Most mechanical devices, regardless of the existence of electrical components, can be viewed as a singular system or a system-of -systems. A gearbox may be seen either as a system for changing gear ratios, as a part of a transmission system, as a part of a drive system or similar. Even if the gearbox is viewed as a singular system it may still be possible to divide it into a functional part, the gears, and an enabling part, the bearings, and if present even to a controlling part, a gear selector. Loss of function in any individual subsystem will probably reduce the overall function of the gearbox, but not necessarily make it inoperable. Should the gearbox be viewed as part of a transmission system the problem becomes more complex making the need for a more sophisticated logic for accurate diagnostic. Damages to peripheral parts of the transmission system might increase the wear on individual gears making their predicted wear inaccurate. In respect, problems originating in the gearbox might lead to increased wear on things like bearings and motors but not necessarily stop the system ability to operate as a whole. In other words, the system or system of systems experiences a loss of function and needs a correct diagnosis to determine a correct cause of action, but the system has not stopped working.

If there was to be an analogy with the human body, a machine would not only be considered either operational or non-operational, it would be considered as either healthy or affected by different levels of function loss. A person suffering from a arthritis in a thumb joint would not be considered non-operational. That person might even be considered quite healthy over-all. The idea is that the amount of wear on a mechanical subsystem, or even a fully mechanical system, cannot be represented by either a 0 (false) or 1 (true), or possibly even a scale e.g. from one to five. Our point in this paper is then also that this is not just a numerical scale, but comes with algebraic structures. It needs to be translated into a much more sophisticated representation to fully represent the complexity of the problem. A classical logical fault-tree consisting of either true or false as possible states of being are not accurate enough even for a small system. A system or system-of-system that could come in question for scrutiny of its dependencies and structures must be equal to a process. There would be little need to perform such task on a static object. In order to sufficiently translate the overall health state of a system or a system-of-system we need to use generalized relations and logical models that allows for order and many-valuedness. This means that the classical fault tree, that uses 0 and 1 to represent operational or non-operational states is replaced by something that is containing enriched information, perhaps in the form of truth values between bot (bottommost truth value) and top (topmost truth value), enabling representation of the operational degree of any given system.

One fundamental aspect of applying any form of algebraic, logic or topological operation on a system of even moderate complexity is to have the means of understanding a real-life-system with its interactions, both internal and external, and to have a tool capable of making a logically coherent visualization of this system. There are a number of established notations available that are more or less widely used to translate different kinds of processes in to structured and ordered representations, or models, of the original. The more complex the system and the more intricate the system-of-system, the higher is the need to find a notation with a rich underlying logic. This is important to allow for design structures to keep relations between the components and data and allow for maintaining both order, many-valuedness and topology.

2 Unstructured and structured information

The simplest form of information is a set X of points $x \epsilon X$. If X is given no structure, and the points x remain unexplained, no mathematics, apart from set theory, can be applied to analyze such 'information'.

Intuitively, we may e.g. say that X_{Co} is a 'set of components' and $x_{crankshaft}$ is a 'component' in X_{Co}, i.e., $x_{crankshaft} \epsilon X_{Co}$. It is then tempting to say that this is more informative than saying $x \epsilon X$, but in fact, mathematics at this point is blind to see any difference between $x_{crankshaft} \epsilon X_C$ and $x \epsilon X$, since $x_{crankshaft}$ is mathematically still just an element and X_{Co} is just a set.

The DSM model (Eppinger and Browning, 2012) is a typical relational model, which informally may define information types, and in the case of DSM roughly divide these types into *components*, *people* and *activities*. Respective types are equipped with underlying and unstructured sets of elements of these types, so that we may add sets X_{Co}, X_{Pe} and X_{Ac}, respectively, of elements representing components, people and activities. However, elements in these sets indeed remain simply as names. Algebraically, logically and topologically we still have very little structure, if any structure at all, except for the possibility to create free algebras, logical signatures with only constants, or trivial topologies.

A typical step and starting point to add structure is to say that "points can be related". We may want to describe how components are related or maybe how components and people are related, and so on. This means we establish relations as subsets

$$R_{CoCo} \subseteq X_{Co} \times X_{Co}$$

and

$$R_{CoPe} \subseteq X_{Co} \times X_{Pe}.$$

We may want to impose various properties on relations, like those for reflexivity, symmetry and transitivity, providing equivalence relations. Such relations divide the set of elements into a set of non-overlapping subsets. Conversely, for any subdivision of a set into a set of non-overlapping subsets we can define a unique equivalence relation that provides that subdivision. Respective subsets are then *per se* unrelated.

The symmetry property essentially means that the relation is unordered, so that asymmetry means that order makes sense. The relation is then more conveniently treated as an order relation, and therefore appears within the realm of lattices and algebras.

Note also how a relation $R \subseteq X \times X$ can be equivalently represented as the mapping

$$\rho : X \times X \to 2$$

where 2 denotes the two-pointed set $\{0,1\}$ $(or \{false, true\})$. The relation has initially no properties, so it may e.g. be asymmetric indicating that the order between components is important. However, order as a structure is not explicitly recognized within the formal notation, and in fact, in the case of DSM, the model comes with very little formal notation.

In design structures, order and many-valuedness are important, but in logic it is an interesting question whether order precedes many-valuedness. If we first extend 2 to Q, a non-commutative quantale, we have a many-valued relation

$$\rho: X \times X \to Q$$

and non-commutativity of the quantale means that aggregations will consider the order among elements in Q, see e.g. (Eklund, Gutiérrez García, Höhle and Kortelainen, 2018). DSM also deals with many-valuedness, but in a rather pragmatic way, and not using algebraic notions or logical formalism to describe it more precisely.

This is clearly seen e.g. in DSM's four types of interactions (spatial, energy, information, and materials), with a 5-scale (-2 … 2) characterizing many-valuedness for each interaction. That 5-scale can be viewed as a quantale, but the relation between respective 5-scales is not algebraically explained in DSM.

Many-valuedness and order is thus poorly explained in DSM, and for the set X must also have a more elaborate structure, otherwise the size of that unstructured set quickly grows to become very large, and application development makes no practical sense. As we indicated before, X cannot be just a *set of* elements. It has to be a *structure of* elements.

As an example, if we only say ‘crankshaft’ as a name for a component in an automotive system-of-systems, ‘crankshaft’ is just a logical constant, but if we include the attributes $attr_1, \dots, attr_n$ attached to a crankshaft it becomes a logical term. Using logical notation, $crankshaft$ is a logical constant (of zero arity), whereas $crankshaft(attr_1, \dots, attr_n)$ is a term, with $crankshaft : s_1 \times \dots \times s_n \to s$ being an operator (of arity n) and s_i, $i = 1, \dots, n$, and s are types (sorts).

In first order logic, $crankshaft(attr_1, \dots, attr_n)$ may be viewed as a term or a predicate. In (Eklund, Höhle and Kortelainen, 2014) terms are clearly separated from sentences, so that $crankshaft(attr_1, \dots, attr_n)$ is an expression (term) rather than a statement or predicate (sentence). Conglomerates of sentences become part of the logical *theory* related with the design structure.

In the simplest case, components are terms, built upon a signature $\Sigma = (S, \Omega)$, where S is the set of types and Ω is the set of operators. The set of all terms (expressions) is then $T_\Sigma X$, where X is a set of variables. The design structure is then

$$\rho : T_\Sigma X \times T_\Sigma X \to Q$$

where order and many-valuedness reside in both components and the valuation of the relation between them. In this situation, T is a functor over the category of sets, so that order and many-valuedness reside in the functor structure. However, as explained in (Eklund, Galán, Helgesson and Kortelainen, 2014), T can more generally be an endofunctor over any monoidal biclosed category, so that order and uncertainty is modeled in the underlying category (metalanguage) rather than in the functor itself.

Further, the relation ρ may be constrained by properties, such as associativity. Applications typically define these properties, as well as the nature of order and many-valuedness.

We can enrich ρ even further, and this makes us realize how DSM without structure is capable of producing applications on a very general level only.

3 Contact relations

People, and people in teams, are obviously differently structured as compared to components and subsystems of components. Relations between and (topological) nearness of people and teams require to be modelled also involving topological notions like neighbourhood, entourage, proximity and nearness. Neighbourhoods of points in topological models originate and abstracts from geometry and metric space models. Entourages in uniform spaces (Weil, 1937) and can intuitively be viewed as two-dimensional or “relational” neighbourhoods. Nearness (Herrlich, 1974) extends proximities (Riesz, 1909), where these models consider proximity of sets rather than points. This brings proximity consideration closer to the notion of *contact relations*.

The mathematical notion of contact has its origin in the so called point-free approach to topology. In recent years, point-free descriptions, i.e., region-based theories of space, in particular, have been a prominent area of research. Traditionally, space has been considered in mathematics by point-based theories such as geometric (e.g. Euclidean geometry) or topological representations (point-set topology) of space. Representing a region by the set of its points might be impossible or at least very inefficient when it comes to computer applications. As an alternative point-free theories of space such as region-based theories can be used to represent space in the context of qualitative spatial reasoning. Using regions instead of points as basic entities accounts more naturally for how humans conceptualize the physical world. For this reason this alternative representation of spatial entities and their relationships has become a prominent area of research within AI and Knowledge Representation. Since the earliest work of de Laguna (deLaguna, 1922) and Whitehead (Whitehead, 1929), mereotopology has been considered for building point-free theories of space. Mereotopology is a combination of the topological notion of connectedness with the mereological notion of parthood. A common mereological approach is to use Boolean algebras modeling the parthood relationship of regions. A Boolean algebra is a set B with two binary operations $\wedge, \vee$, a unary operation * and two constants 0,1 so that the following axioms are satisfied:

$a \vee (b \vee c) = (a \vee b) \vee c$	$a \wedge (b \wedge c) = (a \wedge b) \wedge c$	associativity
$a \vee b = b \vee a$	$a \wedge b = b \wedge a$	commutativity
$a \vee (a \wedge b) = a$	$a \wedge (a \vee b) = a$	absorption
$a \vee 0 = a$	$a \wedge 1 = a$	identity
$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	distributivity
$a \vee a^* = 1$	$a \wedge a^* = 0$	complements

With $a \leq b$ iff $a \wedge b = a$ the induced order on B is defined that immediately generalizes the inclusion of set of points to the abstract elements of the Boolean algebra. A so-called contact relation is often used to model the topological aspect of regions of being in contact. Formally, a contact relation $C \subseteq B \times B$ is a binary relation on B. Most commonly, the following axioms for C are considered:

$C0$	$\neg(0Ca)$	null disconnectedness
$C1$	$a \neq 0 \rightarrow aCa$	reflexivity
$C2$	$aCb \rightarrow bCa$	symmetry
$C3$	aCb and $a \leq c \rightarrow aCc$	compatibility
$C4$	$aC(b \vee c) \rightarrow aCb$ or aCb	summation axiom
$C5$	$C(a) = C(b) \rightarrow a = b$	extensionality
$C6$	aCc or $bCc^* \rightarrow aCb$	interpolation axiom
$C7$	$a \neq 0$ and $a \neq 1 \rightarrow aCa^*$	connection axiom

The first axiom says that no region is in contact with the empty region and $C1$ requires that every non-empty region is in contact to itself. The symmetry axiom makes contact a symmetric relation. This axiom makes perfectly sense in the spatial interpretation. However, if we consider parts of an engine or a system and interpret contact to model the potential influence of a mail function in one part on the other part, this axiom might not be suitable. $C3$ relates the order structure, i.e., the mereological notion, to the notion of contact. The summation axiom states that if a component a is in contact to a component that consists of two parts, then a must be in contact to at least one of the parts. The extensionality property ties the mereological notion to contact. It requires that if to components are contact to the same set of parts, then they are equal. As a consequence the order relation becomes definable in terms of C. The interpolation axiom is an axiom that stems from contact relations obtained by proximity spaces. It is a separation property requiring that two disconnected regions, i.e., two regions that are not in contact, there is a third region disconnected from the first including the second as non-tangential part. Finally, $C7$ requires that every non-trivial region is connected to its complement.

Please note that Boolean contact algebras, i.e., Boolean algebras together with a contact relation satisfying $C1 - C4$, can be represented in topological spaces with the usual definition of contact. In this context the additional axiom correspond to certain properties of the topological space.

4 The Information & Process view of relational structures

In order to translate real world systems into some equivalent representation that can be manipulated and interpreted, some kind of transitional layer is needed. Careful use of BPMN or DMN to capture a real-world process may both preserve and reveal relations between active components in a logically consistent way. Tools like BPMN can be used to make representations of many things and system of systems are just one example outside the business world. Since BPMN and its siblings allows for dependencies like directional flows and relations the addition of weights and values makes them well suited to apply logic to allow for better ways to understand the inner workings of any system of systems, they do however have limitations.

In (Eklund, Johansson, Kortelainen and Salminen, 2017) the logically extended view of DSM was promoted with respect to design structure becoming potentially supported by *information and process* standards as appearing in the OMG (Object Management Group) family of languages and notations, including

- UML (Unified Modeling Language)
- SysML (Systems Modeling language)
- BPMN (Business Process Modeling Notation)
- CMMN (Case Management Model and Notation)
- DMN (Decision Model and Notation)

UML's Structure Diagram is a database model, whereas the Behaviour Diagram in UML is less recognized and used. The Behaviour Diagram in fact is a process model. Further, UML's Behaviour Diagram is part of SysML, which is a process model expanding the process model side of UML. SysML is intended e.g. to support systems-of-systems modeling in engineering and manufacturing. BPMN in OMG should not be confused with value chain models, and and the logic of DMN is basically a propositional logic on a very trivial and basic level. Systems-of-Systems indeed embrace UML, SysML, BPMN, CMMN and DMN, in a variety of combinations.

References

P. Eklund, P., Johansson, M., Kortelainen, J., Salminen, V., 2017. The logic of DSM, In: Understand, Innovate, and Manage your Complex System!, Proceedings of the 19th International DSM Conference (Eds. K. Hölttä-Otto, T. R. Browning, S. D. Eppinger, L. Becerril), Espoo (Finland), 11–13 September 2017, 25-31.

de Laguna, T. (1922). Point, line and surface as sets of solids. The Journal of Philosophy 19, 449–461.

Eklund, P., Galán, M.A., Helgesson, R., Kortelainen, J., 2014. Fuzzy terms, Fuzzy Sets and Systems 256, 211-235.

Eklund, P., Gutiérrez García, J., Höhle, U., Kortelainen, J., 2018. Semigroups in Complete Lattices: Quantales, Modules and Related Topics, Developments in Mathematics 54, Springer, 2018.

Eppinger, S.D., Browning. T.R., 2012. Engineering Systems: Design Matrix Methods and Applications, MIT Press.

Herrlich, H., 1974. A concept of nearness, General Topology and its Applications 5, 191-212.

Riesz, F., 1909. Stetigkeit und abstrakte Mengenlehre, Rom. 4. Math. Kongr. 2: 18–24.

Weil, A., 1937. Sur les espaces à structure uniforme et sur la topologie générale, Act. Sci. Ind. 551, Paris.

Whitehead, A. N, 1929. Process and reality. MacMillan, New York.

Contact: M. Johansson, Umeå University, Department of Computing Science, Umeå, Sweden, +46 90 786 6507, matsj@cs.umu.se

Analyzing Complex Socio-Technical Systems in Technical Product Development Using Structural Metrics

S.C.M. Knippenberg[1], S. Schweigert-Recksiek[2], L. Becerril[2], U. Lindemann[2]

[1]Eindhoven University of Technology, The Netherlands
[2] Technical University of Munich, Germany

Abstract: Structural complexity metrics provide information about the structure of technical and socio-technical systems, represented as networks. However, calculating multiple metrics of a network manually requires a lot of time and effort. Thus, to increase the efficiency in structural complexity management, a tool in Soley Studio is proposed that performs analyses of complex networks automatically. This tool analyses socio-technical systems networks using a set of structural metrics and supports the visualization of the results. Here, three of the structural metrics implemented are presented in depth and applied to a case study of an electrical Formula Student racing car.

Keywords: Structural Complexity Management, Structural Metrics, Graph-based Analyses, Product Development, Communication

1 Introduction

The increasing complexity in product development is inevitably coupled to complexity in engineering design processes and the organization conducting the product development (Sosa et al. 2004; Schweigert et al., 2017). Especially when different departments in the organization need to work together, for example between the design and simulation departments, the growing product and process complexity lead to additional challenges. Therefore, methods of complexity management like matrix-based or graph-based approaches have a long tradition of application in handling complex product development processes and structures (Eppinger & Browning, 2012).

Graph-based approaches gain increasing attraction in the community as the tool landscape is growing. The resulting visualizations are useful for decision making and are arguably in many cases easier for non-experts to understand - compared to matrices (Kissel, 2014).

Furthermore, metrics add a quantitative dimension to the often qualitative results of graph-based analyses. When combining these two techniques – graph-based approaches and metrics – holistic analyses of engineering design processes and collaboration networks can be conducted (Kreimeyer, 2009). However, to apply these analyses in industry consistently, it is necessary to enhance their usability and improve the cost-benefit relationship. Therefore, this paper proposes a metrics toolbox implemented in the graph-based tool Soley Studio. This toolbox contains workflows that calculate structural metrics for analyzing collaboration networks at department interfaces and for estimating the understandability and transparency of the modeled systems.

2 Related Work

The toolbox developed in this work is based on existing structural complexity metrics. This section gives a brief overview of existing work on complexity metrics within technical product development, as well as related work.

The work of Kreimeyer (2009) describes 52 structural metrics that can be applied on complex networks of engineering design processes for providing additional insights. These metrics will generate a practical application by applying structural complexity management on complex engineering networks.

The insights in complex networks provided by the structural metrics can be used for gathering information about existing process models and for structuring new process models consistently (Mathieson and Summers, 2017; Schweigert et al., 2017), as understanding the structure of a system is essential for predicting its behavior (Oehmen et al., 2015). Furthermore, this information about the complex network structure can reduce the risks in the planning of processes through better perception of impacts or changes.

Building on the Goal-Question-Metric approach by Basili et al. (1994), the metrics shown in Table 1 in Section 4.1 have a translation to barriers at the interface of design and simulation departments (cf. Schweigert-Recksiek and Lindemann (2018) for details). While the term metrics is often used in the sense of performance metrics in engineering design (O'Donnell & Duffy, 2005), this paper focuses on structural metrics. The sources for these structural metrics are listed in Table 1 in Section 4.1.

Further metrics, such as “cognitive weight” capture the understandability and user-friendliness of the modelled system (Wang, 2006). Thus, areas within the modelled system that are hard to comprehend can be identified, and for example, be the focus of trainings.

Moreover, computing understandability-related metrics automatically will allow, in future work, to develop a self-optimizing presentation of qualitative analysis results as graphs by displaying the largest amount of information that is still understandable for the human analyst.

3. Methods

For the development of a toolbox for managing complex systems using structural metrics and to validate its working, a case study is performed. The upcoming sections provide information about the dataset on which the case study is conducted and which graph-based tools are used for the implementation of the proposed toolbox.

3.1 Dataset University Racing Eindhoven

To illustrate the application of the toolbox, this paper presents the analysis of a dataset obtained from design documentation of the Formula Student team of the Eindhoven University of Technology; University Racing Eindhoven. Every year the team designs, builds, tests, and races a single-seated formula-style racing car. In 2015, the team built its first four-wheel drive electrical racing car and has already realized its fourth, electric, four-wheel drive racing car from which the dataset is obtained. The goal of the case study behind the dataset was to improve the test steps and integration steps within the development of the racing car.

3.2 Soley Studio

Soley Studio is a commercially available tool for modelling, analyzing and visualizing graph-based data models and allows to modify and develop analysis solutions. Therefore, it is suitable for the determination of structural metrics for a complex network. The data of a network can be visualized in a graph, to which different layouts can be applied. Even though multiple tools for that purpose are available on the market, Soley Studio has been chosen since users are able to program and share tailored analyses workflows.

Furthermore, the software solution is equipped with a multiplicity of library elements for analyzing data, which can be combined and extended by the user for creating a tool with desired functions. These extensions can be created using a programming language that is based on the GrGen.NET documentation (Jakumeit, 2017) for graph modeling, pattern matching, and rewriting. The data that is imported in Soley Studio can be transformed and analyzed based on transformation rules for graph-based models, after which it can be presented as graphs, charts, tables or matrices.

3.3 GrGen.NET

In 2003, the open source GrGen project was established as a response to the demand for a software development tool for analyzing graph-based intermediate representations. As a result, GrGen.NET was developed, which has been developed into a tool for pattern matching and graph rewriting that is applicable for general applications (Jakumeit, 2017). Furthermore, GrGen.NET is used for transforming intuitive and expressive rule-based specifications into efficient .NET code (Jakumeit, 2010).

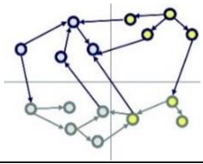
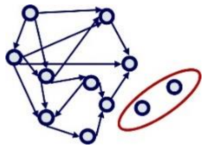
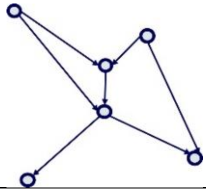
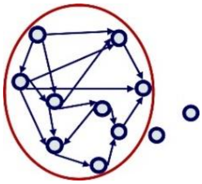
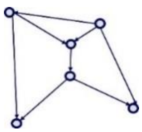
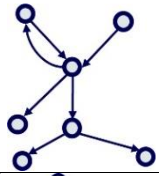
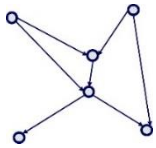
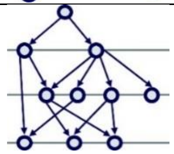
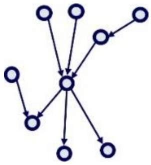
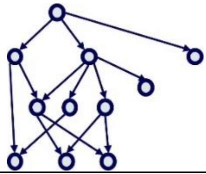
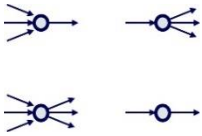
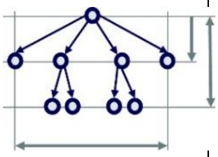
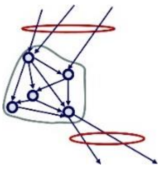
4. Deriving Structural Metrics from Collaboration Graphs

Managing complex systems in technical product development can be performed by deriving structural metrics form collaboration graphs. In this section, an overview of the metrics that are implemented and visualized using Soley Studio is presented, after which three metrics are depicted and applied on the dataset of University Racing Eindhoven.

4.1 Metrics Overview

A selection of fourteen metrics out of the 52 structural metrics as described in the work of Kreimeyer (2009), as presented in Table 1, has been implemented in Soley Studio for analyzing complex networks. From these implemented metrics, three exemplary metrics are expanded in the next sections. These metrics are then applied on the case of University Racing Eindhoven (c.f. Section 3.1).

Table 1. Overview of the fourteen metrics (Kreimeyer, 2009) of which the calculation is implemented in Soley Studio.

Structure	Metric	Structure	Metric
	**1:** Number of Domains (Gruhn & Laue, 2006)		**8:** Number of Unconnected Nodes (Maurer, 2007)
	**2:** Number of Nodes per Domain (Azuma & Mole, 1994; Browning, 2002; Gruhn & Laue 2006)		**9:** Number of Connected Nodes
	**3:** Number of Edges per Domain (Browning, 2001)		**10:** Number of Reachable Nodes (Maurer, 2007 202)
	**4:** Number of Edges per Node (Browning, 2002)		**11:** Height of Hierarchy (Maurer, 2007, p. 218)
	**5:** Outgoing (Activity) and Incoming (Passivity) Edges per Node (Lindemann, 2007)		**12:** Width of Hierarchy (Maurer, 2007; Robertson & Seymour, 1986)
	**6:** Degree Correlation (Nodes) (Ahn et al., 2007; Nikoloski et al., 2005)		**13:** Snowball Factor (Loch et al., 2003)
	**7:** Fan Criticality (Gruhn & Laue, 2006)		**14:** Cognitive Weight per Domain (McQuaid, 1997; Shao & Wang, 2003; Wang, 2006)

4.2 Activity and Passivity

The first metric describes the number of outgoing (activity) and incoming (passivity) edges per node. The output is a list of values for activity and passivity for each of the nodes within the domain. The results are visualized using an influence portfolio (Lindemann et al., 2009). It can be used to classify the intensity of changes in the network acting on a certain node. Furthermore, the nodes with the highest relevance within the network can be identified.

For determining the metric, standard library elements in Soley Studio are used for determining the activity and passivity for each of the nodes in the network. Using both the activity and passivity, the criticality of each of the elements can be calculated using Equation 1 (Lindemann et al., 2009).

$$Criticality = Activity \cdot Passivity \qquad (1)$$

A high criticality of an element indicates a high number of indirect dependencies. *Critical* elements are strongly interlinked within the network and therefore have a high influence on the overall system behavior. Changes to these critical elements can influence large parts of the network and should therefore be avoided when radical changes are not desirable.

Besides the *critical* elements, the elements with a criticality low value are indicated as *inert*. These elements are weakly interlinked in the network and changes would not affect a large number of other elements.

4.3 Snowball Factor

The snowball factor, as presented in Figure 2, describes a measure for the spreading of information or errors within a network and is the sum of the product of the height and width of the hierarchy of the considered network. The height of the hierarchy is defined as the number of levels that are present in the tree structure of a network and is determined level by level. The width of the hierarchy is determined level by level and is defined as the number of leaf nodes for each of the levels of a tree structure in a network. Leaf nodes are located at the end of the hierarchy and have incoming edges only. When a node is accessed more than once from different levels, the lowest level is used for the computing.

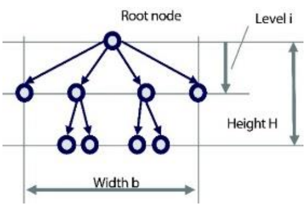


Figure 2. Snowball factor, spreading of information or errors within a network (Kreimeyer, 2009).

For each of the levels, the snowball factor is weighted with the inverse of the length of the shortest path to the root node. The root node of which the snowball factor is determined, is defined as a node without any incoming edges. Therefore, a root node with both incoming and outgoing edges cannot be defined as a root node for calculating the snowball factor. Furthermore, passive root nodes cannot be defined as root nodes for determining the snowball factor of a network. These nodes are defined as nodes with incoming edges for retrieving data from other nodes of the network. When this condition for the root node is met, the snowball factor is determined by calculating the sum of the product of both the height and width (per level) of the hierarchy, starting from a defined root node. In this calculation, each of the levels of the hierarchy should be weighted with the inverse of the shortest path length to the root node, as presented in Equation 2 (Kreimeyer, 2009). In this equation, H is the highest level that is taken into account, i represents the current level for determining the snowball factor, and b stands for the width of level i of the network.

$$Snowball\ factor = \sum_{i=1}^{H} \frac{b_i \cdot H}{i} \qquad (2)$$

From this equation, it can be noted that the shortest path to the root node is equal to the difference between the total height of the hierarchy and the height of the specific level.

4.4 Cognitive Weight

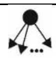
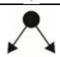
To describe the human ability to understand both particular parts of the network and how a network is structured, a metric for describing the cognitive weight is defined by Wang (2006). This metric represents the sum of the cognitive weight of each individual node that is part of the network.

The calculation of Metric 14 is performed in two different ways, since the metric can be defined slightly different. For the first manner, as shown in Equation 3, the highest cognitive weight for each of the nodes is assigned if multiple cognitive apply.

$$CW_j = max(e_j\ ,3) \qquad (3)$$

In this equation, CW_j is the cognitive weight of node j and e_j is the number of outgoing edges in the assessed network structure. Afterwards, Metric 14 is determined by calculating the sum of all nodes in the network. In Table 2, an overview of the cognitive weight for different structures of the network is presented.

Table 2: Overview of the cognitive weight for different structures within the network.

Structure	Weight	Structure	Weight
	1		3
	2		3

Evaluating the information as presented in Table 2 (Wang, 2006) may lead to a possible issue calculating the cognitive weight, as described above. Therefore, a second method is introduced for which the cognitive weights of the individual nodes are multiplied, if more than one structure applies, which is presented in Equation 4.

$$CW_k = max(e_k\ ,\ 3) \cdot l_k \cdot 3 \qquad (4)$$

Here, CW_k is the cognitive weight of node k, e_k is the number of outgoing edges, and l_k is the number of loops in the network, multiplied by *3* for assigning its cognitive weight.

4.5 Case Study

To apply the developed workflows, the 36 main components of the University Racing Eindhoven dataset, and their interdependencies are modelled.

The results of the first analysis (activity and passivity, c.f. section 4.2) are depicted in the influence portfolio in Figure 3. In this figure, the number of incoming and outgoing edges per node are visualized. The elements in the first quadrant represent the most critical components in the network in red. The passive elements are displayed in yellow in the second quadrant. The blue, active elements in quadrant 4. Changes in the inert elements in the third quadrant, indicated with a green color, will have a minor effect on the network and its structure. Furthermore, the diameter represents the criticality of a component (c.f. Lindemann et al., 2009).

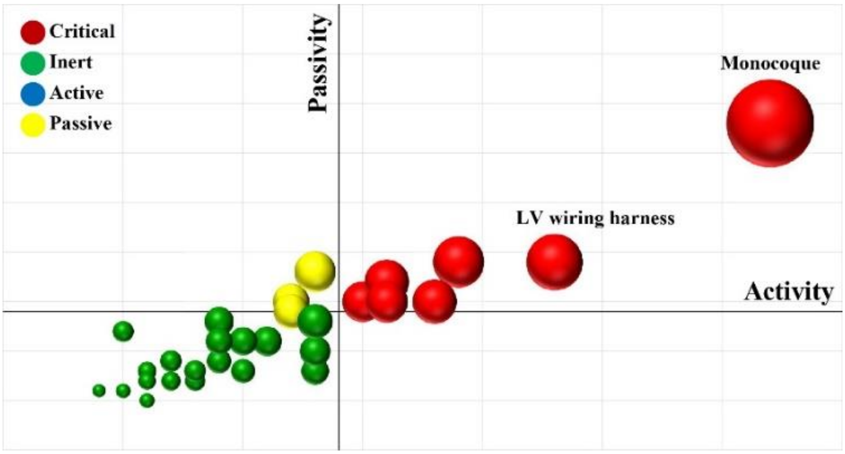


Figure 3. Influence portfolio of the components from the dataset of University Racing Eindhoven and their interdependencies.

As shown in Figure 3, in the case of <University Racing Eindhoven, the most critical components are the Monocoque and the low voltage (LV) wiring harness. All parts are connected via the body (Monocoque) of the racing car and multiple components are powered, controlled by or communicating over the LV wiring harness. Thus, the results of the influence portfolio are deemed plausible.

The metric that describes the snowball factor (section 4.3) only exists for root nodes of a structure as shown in Figure 2. Thus, no metrics can be calculated for elements in the whole network of the case study, since the network does not contain root nodes and is highly interconnected. Nevertheless, the snowball factor can be calculated for isolated groups of edges and nodes.

For determining the cognitive weight of the network of University Racing Eindhoven and indicating the difference between the two described methods for determining the metric, as presented in Equations 3 and 4, the components of the dataset are divided into five domains. In Figure 4, the results of the calculation of the metric for both methods are presented. The left graph represents the cognitive weight when applying the "Highest value" method and the right graph shows the results for the "Multiplied" method.

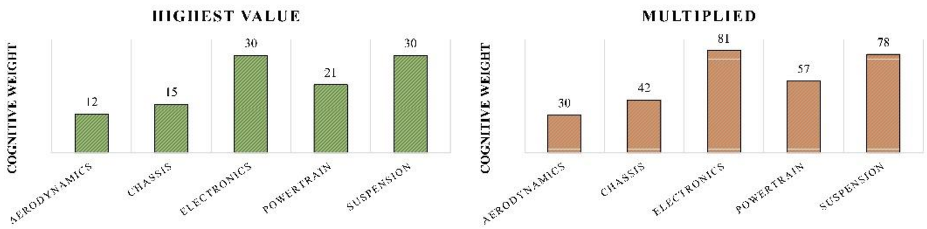


Figure 4. Cognitive weight for each of the domains, determined using the "Highest value" method on the left and using the "Multiplied" method on the right.

Here, the difference between both methods can be identified. Where the value of the cognitive weight for the domains Electronics and Suspension is equal for the "Highest value" method, a difference can be seen for the "Multiplied" method. An explanation is that multiplying the cognitive weights for more complex structures results in higher values. In the same situation, the other method assigns the value of the most complex structure as cognitive weight. As a consequence, this method does not penalize all complex structures where the multiplying method takes every composition into account.

5. Conclusion and Outlook

This paper presents the implementation of a set of metrics using Soley Studio. The goals of the implementation are a) to improve the usability and cost-benefit of Structural Complexity management analyses in practice; and b) to create a metrics "library" that fosters comparability among and analyses of different datasets, therefore improving reproducibility.

Using the in Soley Studio implemented tool, the user is able to obtain additional insight into extensive datasets by applying structural complexity management. The toolbox or library developed contains fourteen metrics that facilitate a range of insights regarding a technical system being developed and the socio-technical system that develops it. One application we address is enhancing the communication and collaboration between different departments, e.g. to indicate which barriers exist in certain collaboration networks and to identify recommendations for improvement measures to overcome the barriers.

Moreover in this paper, we focus on three metrics (activity and passivity, snowball factor, and cognitive weight), which are explained in detail in Sections 4.2 to 4.4 and applied to a case study from the University Racing Eindhoven (Section 4.5). Based on the metrics applied to this dataset, the following two insights about the system can be drawn:

- The Monocoque is clearly the most critical part of the architecture. Thus, the person responsible for its development has to be integrated thoroughly in the overall information flow of the project.
- Due to the high cognitive weight of the networks concerning the domains electronics and suspension, these two areas are prone for the analysis with structural metrics, as a conclusion cannot be drawn just from visual analyses.

The main challenge in this work was the fact that many metrics are not defined very clearly in literature leading to different implementation possibilities. This contribution overcomes this obstacle by sharpening the definitions during their implementation. The industrial benefit of the presented metrics library lies in the possibility of quickly analyzing complex collaboration structures in a standardized way.

In future work, additional metric calculations can be implemented to obtain further insights when applying structural complexity management. To identify additional metrics that need to be implemented, additional datasets with different structures can be analyzed. In addition, the toolbox is currently applied to student teams in research projects to test their usability and will be used in industrial case studies in the near future. This will provide insights on the usefulness of the conclusions to be drawn from them as well as their industrial benefit.

6. Acknowledgements

We would like to thank University Racing Eindhoven, the Formula Student team of the Eindhoven University of Technology, for providing information and data for this research. Furthermore, we would like to thank Pascal Etman, Associate Professor of Mechanical Engineering in the Control Systems Technology group at the Eindhoven University of Technology, for supervising the research project.

References

Amabile, T.M., 1996a. Creativity in context. Westview Press, New York.

Ahn, Y.-Y., Han, S., Kwak, H., Moon, S., Jeong, H., 2007. Analysis of Topological Characteristics of Huge Online Social Networking Services. In: Proceedings of the 16th International World Wide Web Conference, Banff, Alberta, Canada, 08.-12-05.2007. New York, NY: ACM 2007, pp. 835-844. ISBN: 978-1-59593-654-7.

Azuma, M., Mole, D., 1994. Software Management Practice and Metrics in the European Community and Japan: Some Results of a Survey. Journal of Systems and Software 26, pp. 5-18.

Basili, V. R.; Caldiera, G.; Rombach, H. D.: Goal Question Metric Paradigm. In: Marciniak, J. J. (Ed.): Encyclopedia of Software Engineering. New York, NY: Wiley 1994, pp. 528-532. ISBN: 0-471-54004-8.

Browning, T.R., 2002. Process Integration Using the Design Structure Matrix. Journal of Systems Engineering 5, pp. 180-193.

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, pp. 292-306.

Eppinger, S.D., Browning, T.R., 2012. Design Structure Matrix Methods and Applications. The MIT Press, Cambridge, Massachusetts.

Gruhn, V., Laue, R., 2006. Komplexitätsmetriken für Geschäftsprozessmodelle. In: Mayr, H.C. et al. (Eds.): Proceedings of the Modellierung 2006, Innsbruck, Austria, 22.-24.03.2006. Bonn: Gesellschaft für Informatik, GI 2006, pp. 289-292. ISBN: 3-88579-176-5.

Helms, B., Shea, K., 2010. GrGen.NET: Object-Oriented Concepts for Computational Design Synthesis. 11th International Deisgn Conference, Dubrovnik, Croatia, pp. 1333-1342.

Jakumeit, E., Buckwald, S., and Kroll, M., 2010. GrGen.NET: The Expressive, Convenient and Fast Graph Rewrite System. International Journal on Software Tools for Technology Transfer, vol. 12, pp. 263-271.

Jakumeit, E., Blomer, J., and Geiß, R., 2017. The GrGen.NET User Manual. Universität Karlsruhe, Fakultät für Informatik, Institute für Programmstrukturen und Datenorganisation, Lehrstuhl Prof. Goos. 1.4 ed.

Kissel, M.P., 2014. Mustererkennung in Komplexen Produktportfolios. PhD Thesis, Technical University of Munich.

Kreimeyer, M.F., 2009. A Structural Measurement System for Engineering Design Processes. PhD Thesis, Technical University of Munich, Verlag Dr. Hut, Munich.

Lindemann, U., Maurer, M., Braun, T., 2009. Structural Complexity Management: An Approach for the Field of Product Design. Springer-Verlag, Berlin Heidelberg.

Lindemann, U., 2007. Methodische Entwicklung Technischer Produkte. Berlin: Springer 2007. ISBN: 3-540-37435-3.

Loch, C., Mihm, J., Huchzermeier, A., 2003. Concurrent Engineering and Design Oscillations in Complex Engineering Projects. Concurrent Engineering: Research and Applications 11, pp. 733-750.

Mathieson, J.L., Summers, J.D., 2017. A Protocol for Modeling and Tracking Engineering Design Process Through Structural Complexity Metrics Applied Against Communication Networks. Concurrent Engineering, Vol. 2, Issue 2, pp. 108-122, June 2017.

Maurer, M., 2007. Structural Awareness in Complex Product Design. Dissertation, Technische Universität München. München: Dr. Hut 2007. ISBN: 978-3-89963-632-1.

McQuaid, P.A., 1997. The Profile Metric and Software Quality. In: Proceedings of the International Conference on Software Quality. Montgomery, AL, 06.-08.10.1997. Milwaukee, WI: American Society for Quality 1997, pp. 245-252.

Nikoloski, Z., Deo, N., Kucera, L., 2005. Degree-correlation of Scale-free Graphs. In: Felsner, S. (Ed.): Proceedings of the 2005 European Conference on Combinatorics, Graph Theory and Applications, EuroComb'05, Berlin, 05.-09.09.2005. Discrete Mathematics & Theoretical Computer Science, DMTCS Proceedings Volume AE (2005), pp. 239-244.

O'Donnell, F. J.; Duffy, A. H. B.: Design Performance. London: Springer 2005. ISBN: 1-85233-889-X.

Oehmen, J., Thuesen, C., Parraguez, P. & Geraldi, J. (2015). *Complexity Management for Projects, Programmes, and Portfolios: An Engineering Systems Perspective*: Project Management Institute, PMI.

Robertson, N., Seymour, P.D., 1986. Graph Minors. II. Algorithmic Aspects of Tree-width. Journal of Algorithms 7, pp. 309-322.

Schweigert, S., Luft, T., Wartzack, S., Lindemann, U., 2017. Combination of Matrix-based and Graph-based Modeling for Product and Organizational Structures. Proceedings of the 19th International DSM Conference Espoo (Finland), 11–13 September 2017, 2017.

Schweigert-Recksiek, S., Lindemann (2018). Improvement Opportunities for the Collaboration of Design and Simulation Departments – An Interview Study. DESIGN 2018 16th International Design Conference, Dubrovnik, Croatia.

Shao, J., Wang, Y., 2003. A New Measure of Software Complexity Based on Cognitive Weights. Canadian Journal of Electrical and Computer Engineering 28, pp. 69-74.

Sosa, M.E., Eppinger, S.D., Rowles, C.M., 2004. The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. Management Science, 50 (12), 1674-1689.

Wang, Y., 2006. Cognitive Complexity of Software and its Measurement. 5th IEEE International Conference on Cognitive Informatics. Beijing, China, pp. 226-235.

Contact: S.C.M. Knippenberg, Eindhoven University of Technology, Department of Mechanical Engineering, Den Dolech 2, PO Box 513, Eindhoven, The Netherlands, s.c.m.knippenberg@student.tue.nl

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Influence Profile of Wastewater Chain in Amsterdam: towards resilient system for Phosphorus Recovery & Valorisation

M. V. Amosov[1], L. Zlatanovic[2], J. P. van der Hoek[2,3]

[1]Organic Village, The Netherlands
[2]Technical University Delft, The Netherlands
[3]Waternet, Amsterdam, The Netherlands

Abstract: The wastewater system of Amsterdam offers an opportunity to recover phosphorus, and contribute to circular economy. However, it remains unclear where to intervene in system to maximize recovery and valorisation in a resilient and feasible way. The Design Structure Matrix method was tested to define the system architecture from Food-Water-Energy nexus perspective. Physical, phosphorus, and ownership dependencies between Infrastructure, Stakeholder, Resource and Cleantech domains (elements) of the wastewater system in Amsterdam are analyzed in a Multi-Domain Matrix model. Change Propagation Indicator quantified critical elements, and emergent changes. An Influence Profile unveiled four levels of system leverage: household, neighborhood, city-block, region. The stakeholders can engage into optimizations at each level, to generate individual and shared benefits. Hybrid infrastructure, plug&play solutions and modular approach to cleantech will harness up to 100% of phosphorus available. The method proved to be an effective tool for analysing complexity and engineering resilient solutions for the circular economy.

Keywords: wastewater, phosphorus, nexus, DSM, design, resilience, Amsterdam.

1. Introduction

The role of cities in the Circular Economy is humongous. On one hand, cities account for more than 67% of the global greenhouse gas emissions, (IEA, 2008) and consume up to 80% of global resources (Metabolic, 2017). On the other hand, they offer opportunities for climate-neutral, self-sufficient and sustainable living from waste (AMS, 2016). However, most opportunities remain hidden behind the complex interactions of various types of systems (e.g. stakeholders, infrastructure, policies). Unveiling this complexity is a necessary task in optimizing cities towards resilient and healthy living. Water cycle plays a key role in the transition to the Circular Economy (Henriquez et.al., 2017); specifically, wastewater (WW) – as it carries various materials (e.g. cellulose, phosphorus (P), nitrogen) and energy, which could be reused in local and regional economy (Agudelo et.al, 2012). The WW system could cover up to 100% of energy (E), 80% of water (W) and 60% of nutrients (N) demand nationally (van der Hoek et. al., 2017), if changes to existing structure of WW cycle would be applied and managed across domains of the Amsterdam Metropolitan Area (AMA) almost simultaneously, from utility to user (Roefs et. al., 2017). However, the WW system adopts complexity of a city in terms of distribution of infrastructural, governmental, cleantech and resource interdependencies and assets in space, time, quality.

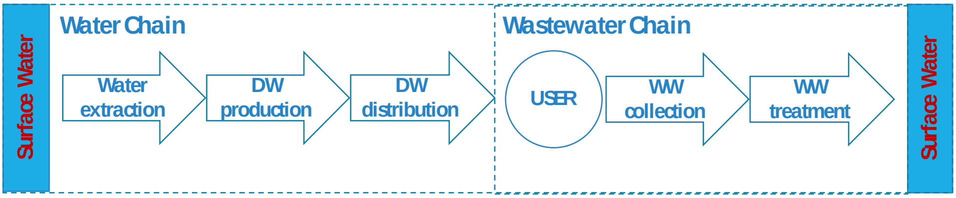


Figure 1. Simple perspective on WW system at AMA (adopted from van der Hoek et.al., 2017)

P is a raw material critical to the European Union (EU), and has strategic value with respect to its recovery (EC, 2017): about 60% of P is located in Morocco, with estimated depletion in 50-300 years (Schoumans et.al., 2015). Annually EU imports 220 tton of P in various products and raw materials; and exports 220 tton as waste. In Amsterdam, up to 60% of P is in WW chain (van der Hoek et.al., 2017), which is mainly withdrawn from the local reuse cycle. P enters WW in a form of detergents, urine, feces, cooking waste, and is delivered by sewers and trucks to WW treatment plants (WWTP); where sludge is incinerated, and the effluent is discharged to the surface water. At present, part of the P (<15%) is recovered through struvite precipitation. P recovery is not just a 'global challenge', but a solution to a local problem: P causes clogging of the infrastructure (pipes and equipment). Recovery of P prevents uncontrolled loss of P and reduces operation and maintenance costs for WWTPs up to EUR 15mln/year. The challenge for Waternet, the water utility of Amsterdam and surroundings, is to increase P recovery. 100% P Recovery & Valorisation (R&V) is of high priority for local stakeholders and a national security. However, it remains unclear to Waternet and linked stakeholders: where to intervene first to recover P or scale-up the pilots in most feasible and resilient way; who is responsible and how benefits are distributed; what should be optimized in a city to 'unlock' the potential; which changes have the most influence etc.

In order to advance decision-making process on the topic of P R&V, and to aid the needs of stakeholders, the study will answer the main research question:

Where to intervene in the architecture around WW chain to recover up to 100% of P in a way that supports the transition of Amsterdam towards resilience?

The main research question can be split up into four sub-questions:

1. What is the definition of the WW chain architecture in Amsterdam?
2. What are physical, P, ownership dependencies of WW elements?
3. Where in the WW chain are the elements critical for change management?
4. How it is possible to recover and valorise 100% of P at AMA?

2. Materials and Methods

2.1 Introduction

The Design Structure Matrix (DSM) is selected as a tool to structure knowledge about complex system into a simple overview of a *system architecture*. It is selected as an effective measure to study changes in the system, such as inflicted by risks (e.g. climate change, population), cleantech (e.g. P recovery) or policies (e.g. EU list of critical raw materials). A case of P R&V from WW system in the AMA is selected to test DSM for studies on potentials for R&V of other resources systematically and in participatory way.

2.2 Case

The AMA is selected as the main System Boundary that included sub-systems physically and organizationally linked to the WW chain – to represent the P propagation from sources to sinks. The final case is a physically connected infrastructure bounded by ownership to the stakeholders that together operate the life-cycle of P through WW and AMA. Where various infrastructural *products* shape the specific WW *process* which is *organized* by a number of stakeholders. P is a *product* that flows through this *organization*.

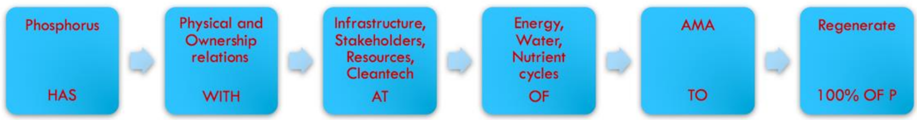


Figure 2. Desired P propagation via established system boundaries

From source to sink P is linked by physical coupling of infrastructure elements (e.g. pipes), including W, Food, and E sectors, that are owned by various actors in the chain, who together influence quality and quantity of P in WW chain, and the cycle of P at AMA. In Amsterdam, there are 1.2 million customers producing 125 million m^3/year of WW and 591.7 tons of P (vd Hoek et.al., 2017). 4000 km of sewers are managed by 20 municipalities. 12 WWTPs are managed by Waternet. WWTP West treats 80% of sludge produced in the AMA and imports additional 179.4 tons. 4200 ha of nature resources are managed by Waternet and regulated by EU. 58.9 tons of P are discharged to the surface water from WWTP, and 598.6 tons are incinerated. Currently, there are four cleantech projects for P recovery in Amsterdam. A micro-scale (house) system at De Ceuvel which generates (theoretically) 50 liters of P a year. A large-scale (street) system at 'Heineken Experience', which generates around 100 tons. A large-scale (neighborhood) system at Buiksloterham that generates (theoretically) 30 tons a year. A large-scale (city) system at WWTP West that generates 500 tons, with estimated potential of 1000 tons. These cleantech produce P for N sectors.

2.3 Design Structure Matrix method

DSM is an <u>*nxn*</u>, square matrix containing nodes and relations within a single domain. Current study adapted an approach of Eppinger et.al. (2012) to create Multi-Domain Matrix (MDM) model (process, product, and organization DSMs). MDM is an <u>*mxn*</u> rectangular matrix containing nodes and relations across 4 domains, where the rows represent one domain and the columns represent another domain. Steps to make DSM are:

1. Decompose: break system's categories down into its constituent elements/nodes.
2. Identify: document the relationships among the system's elements.
3. Analyze: rearrange elements / relationships to understand structural patterns.

4. Display: create a DSM/DMM model, and highlight important features.
5. Improve: through iterations enhance the accuracy and the richness of model.
6. Model: select Variables, establish rules and plot the variables.
7. Evaluate: define Change Propagation Indicators; Critical elements.
8. Design: group elements by score into Influence profile, engineer strategies.
9. Validate: set-up expert meetings to align on terminology, model, and gaps.

Steps 1-9 were repeated 7 times to reach desired level of details in DSMs, and final MDM.

For example, internal components of a house system (site, skin, structure, services, space and stuff (adopted from Brand, 1994)) were decomposed, recorded and characterized in a *House Product DSM* (adapted from Eppinger et.al., 2012), that is arranged by hierarchies of WW, E and N sub-systems (and processes). Physical coupling of all components in a house resembled the system boundary of P flow, owned by the user. High-grade *Product DSMs* were created for WWTP West, household, cleantech pilots, low-grade – for other systems. Multi-Domain Matrix integrated all DSMs, as in Fig.3.

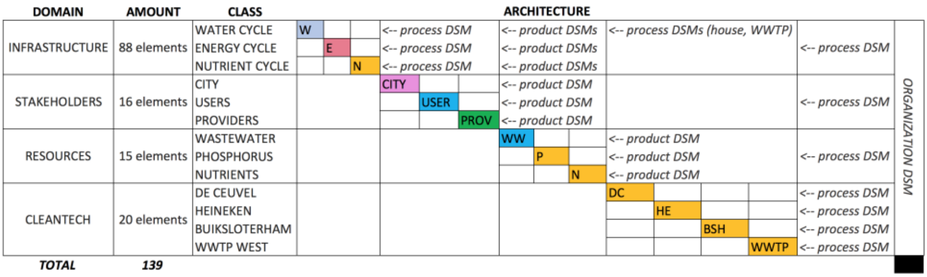


Figure 3. Conceptual design of the final MDM model

Fig. 3 shows simplified view of the final MDM model, and how different products and services are linked and looped to each other. The model is further used to plot physical (spatial), P (material) and ownership (information) dependencies. In order to perform evaluation of the variables between the elements, the basic rule is introduced:

Table 1. The basic Rule

Parameter	Variable	Specification
Spatial	Physical coupling	Is connected = 1; not = 0
Material	Phosphorus coupling	Is present = 1; not = 0
Information	Ownership coupling	Is owning = 1; not = 0

Each variable is plotted into MDM model, and assigned and value ‘1’ or ‘0’. In step 7, the change propagation indicators (CPI) are calculated using these values. Change propagation indicates how a change to one element of a process results in additional changes either within or different parts of the design, whether or not the change initiator is aware of propagation consequences. CPI is calculated for each element by summing all incoming variables ($\Sigma CPI_{2,\ IN}$); and outgoing variables ($\Sigma CPI_{1,\ OUT}$), and then calculated by deducting incoming variables from outgoing (see in Fig. 4):

$$\Delta CPI = \Sigma CPI_{1,\ OUT} - \Sigma CPI_{2,\ IN} \qquad (1)$$

Fig. 4.a shows a conceptual design of the change network diagram type created for the case-study to navigate the population of model with data. Fig. 4.b shows the resulting DSM model and analysis of the CPI with critical elements indicated as Multipliers (M), Carriers (C), and Absorbers (A) of change. ΔX is the external change driver that is leading to a risk or change in component A. It can be seen that element A is changed as a result of an external change, signified by ΔX. This change driver can be related to policy, markets, customer demands or similar changes that take place in the context in which the technology / element has to function. The change to a system component A can be considered the initial change or an innovation as a response to the change driver. This innovation is however not isolated; it rather requires more changes to the system. In the generic example of Fig. 4.a the initial change to component A, propagates the change to the component B, C and E, which themselves propagate change from B – N D, F; E – N F and C – N B, E. This type of change is called emerging change (Eckert et al., 2004).

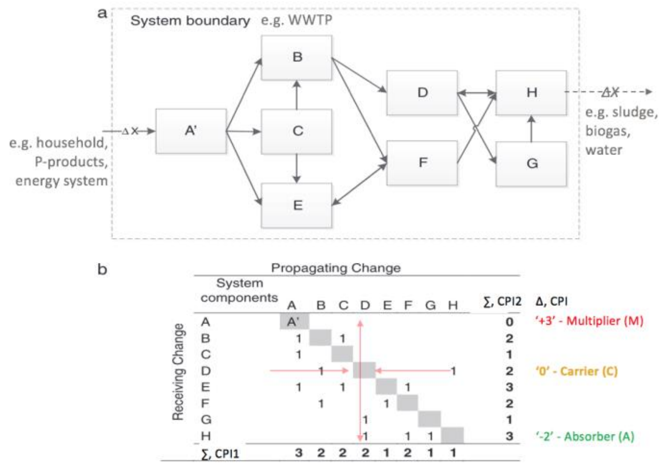


Figure 4. Analytical framework for CPI (graphics adopted from Spiller 2017)

The $\Sigma CPI_{2,\ IN}$ informs on the vulnerability of elements to a change driver. The highest values are considered as points for optimization. E.g. to increase flexibility or/and reduce incoming dependencies of an element. ΔCPI represents an influence of an element on the system when a sequence of changes occurs simultaneously. '+' value means that the element is M, 0-value means that the element is C, the negative value - A. This classification is of value as it draws attention to the key systems elements. "Multipliers are prime candidates for incorporating flexibility. These are elements that, as more changes are added, make the system harder to change". M - propagate more changes than receive. Carriers – propagate as many, absorbers – receive more. One must investigate elements connected to M elements to understand the nature of change. These elements as well might require flexibility to reduce or even eliminate change propagation altogether.

In step 8, by applying a *clustering algorithm*, the elements with the highest scores (physical, P and ownership) and closest locations were grouped into an *Influence Profile* (IP). IP informs on closely related networks of elements that can be integrated or optimized, so to tackle the problems emerging from a complex system. IP is used in further steps to design and evaluate strategies for systemic interventions. Validation step was performed by comparing the data from various sources; including meetings with experts from academia, government and business, and workshops with mixed groups of students and stakeholders in Amsterdam and Singapore. Over 100 publications, technical documents, presentations, brochures were used as the main sources of data.

3. Results and Discussions

139 elements across 4 domains related to P R&V from WW were integrated into the MDM model, and analyzed. MDM provided an insight into distribution and concentration of P flow, structures of technologies and sub-systems, owners, and relations in and outside WW chain – providing an answer to the sub-questions 1 and 2. As a result, the architecture of the WW chain is defined as a tightly coupled physical hierarchical system that cascades the flows of P-products from Sources (e.g. house) to Sinks (e.g. nature) through different levels of ownership; dependent on drinking W, E, food, and solid waste products and services (domains), which together affect quality and quantity of P in the WW and the AMA. Physical, P and ownership dependencies revealed patterns in the design of AMA.

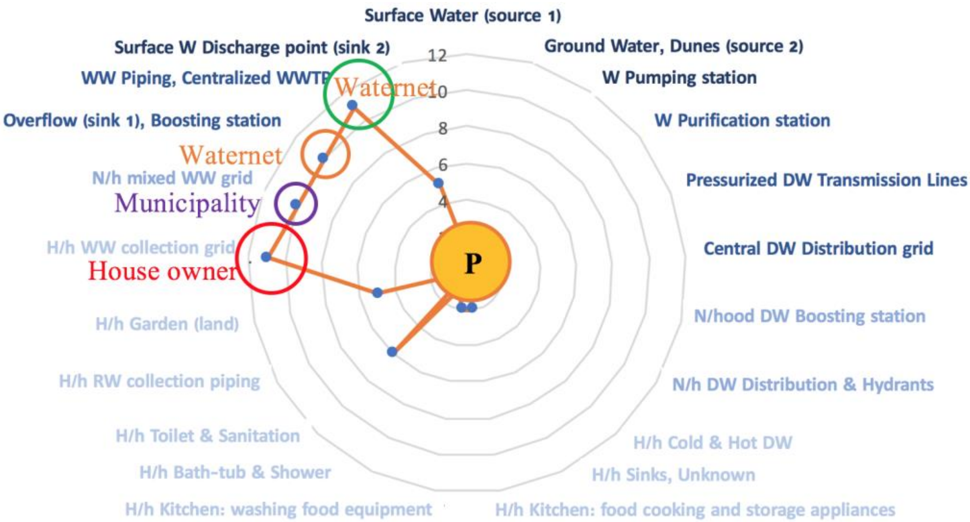


Figure 5. Points for P R&V at Water Cycle of Amsterdam

Fig.5 presents WC of AMA system: where to recover P, and whom to engage. It shows 4 points where P can be recovered: at house, street, neighborhood and city levels by house owners, municipalities, Waternet. Change propagation analysis showed Critical Elements.

Table 2. Change Propagation table: selection of top Critical Elements

Category	Top Critical Elements	ΔCPI
Infrastructure	E Distribution Infrastructure	M
	Hot & Cold Drinking W Interfaces	M
	Toilet & Sanitation	A
	Gravity-based WW Sewer Lines	A
	WW Boosting Station	C
	Pressurized Centralized WW Sewer Line	A
	WWTP	C
	Nature-based WW discharge point (sink2)	A
Stakeholders	Citizens (house owner)	M
	Municipality	M
	Solid Waste Management Utility	M
	Waternet	A
	Cleantech providers	A
Resources	Drinking W	M
	Rain W	M
	Electricity	M
	WW	C
	Kitchen waste	C
	Sludge	A
Cleantech	De Ceuvel, 'Struvitje' (house)	C
	Buiksloterham, 'Resource Station' (hood)	C
	WWTP West, 'Fosvaatje' (city)	C

The Table 2 rates the candidates for the change management in the current design. It predicts the roles of elements, and how they will act as a change driver or receiver in established physical constraints (the design). Given certain changes (e.g. P-recovery), one can predict how change will propagate across the design, through direct and indirect dependencies. For example, *Toilet & Sanitation* is *absorber* of change, owned by *citizen*, with highest concentration of *P*. P-recovery will require less changes to the *sanitation*, however, it will impact e.g. *drinking W interfaces* that are *multipliers*, which will propagate to other *infrastructures* at household and outside (e.g. to *WWTP West,* which is *C, etc.*). In practice, e.g. application of a *vacuum sanitation* would result with higher efficiency of WW transportation system, and reduce leakage, but cost more energy. In this way, an overview of systemic transformations at each domain is derived, answering the research sub-question 3; and allowing further interpretations.

Selected critical elements are grouped into an Influence Profile (IP) of the WW chain based on values of ΔCPI, direct and indirect dependencies (Fig. 6). IP shows points of intervention distributed across four levels of the WW chain: 1 - household, 2- street and neighborhood (combined), 3 - city and 4 - region. Fig. 6 shows 4 leverage points across a selection of infrastructure and stakeholders where interventions and change management strategies can be applied most effectively in individual and/or integrated manner so to reduce the emerging changes at each level (separately) and/or to tune them across all levels. For example, at a *household level* *sanitation*, *drinking & hot W* infrastructure, *E system*, *kitchen (waste) services* can be integrated into a (semi) self-sufficient system for P-recovery, beneficial for other systems inside and outside the house. However, the WW system (e.g. *sewers*) itself acts as a centralized platform, which if made flexible, can adopt (absorb) P-recovery techniques (and emerging changes) from each level of intervention. In practice, a combination of plug&play solution at house, modular solution at neighborhood and a platform solution at city levels could unlock a *hybrid* approach that would provide required flexibility and resilience to the established system design. E.g. low-density area can adopt plug&play solution at house level; as the area grows – modular street-level systems could replace the latter, and if necessary – connect to the sewers or advanced natural environments for post-treatment. The current WW chain can act as a platform to carry and absorb changes around recovery and valorisation of P, if the critical elements (simultaneously) tuned-in towards each other across entire design Such approach to infrastructure and its interfaces would allow flexibility in transition towards 100% P-regeneration via range of feasible solutions that are designed for change. These physical interventions will require changes among relevant owners, and coordination. The IP is a roadmap that answers the main research question, and provides strategic insight for further interpretations, in-depth analysis, and engineering scenarios for resilient P R&V at AMA. The method tested in this study provides guidelines for further research and development.

- The DSM method application adapted from Eppinger et.al. (2012), allowed design of a model about the case from *0-knowledge* to a high definition MDM. DSM and MDM are also applied in cases, such as: *NASA Mars Mission*, *Intel*.
- The MDM model aggregated knowledge and data about WW at AMA to a high level of details, and showed similarities with DSM model of Spiller (2017). However, both WW system and AMA were not explored to an extent as in MDM, which, in fact, can adopt DSM analytics of Spiller (2017) to make better insights.
- The IP of the WW chain, showed similar results as studies of Roefs et. al. (2017) and van der Hoek et.al. (2017). The IP provided additional perspective: how a *hybrid approach* to WW infrastructure domain within context of current AMA architecture can be integrated and leveraged for feasible transition to a resilience.
- The study utilized 1 DSM application out of 100s available (www.dsmweb.org).

4. Recommendations and Further Research

The DSM method is worth further exploration in the field of resilient city systems engineering. It provides a concise overview of a complex system, and a plan for engineering resilient solutions. It also serves as guidelines for participatory research and decision-making. Fundamental nature of DSM – mathematics and graph theory – provide vast opportunities for scale-up of this line of research, especially combined with digital solutions and automation. This method is recommended for structuring circular projects.

However, current MDM data-model is difficult to manage manually. Digital solution would allow automation of data visualization and analysis. As a result, research coverage could be enhanced and shared with other researchers and decision-makers in a user-friendly way. Digital environment would allow application of DSM methods and algorithms, such as *sequencing*, *clustering*, *banding*, *tearing*, *coupling*, *sensitivity* and *network-based* analysis in order to create more innovative insight and a digital framework (e.g. engineering system matrix) for integration of R&D on circular economy around P R&V and other topics. By plotting intended interventions, such as P-recovery or policy, at each level of leverage, we can further design and test the vision for maximum P R&V. More specific strategies and areas of research can be shaped (see Fig. 6). To make such design work, it is necessary to look deeper into content of this roadmap at each level. New products and services can be engineered across Food-Energy-Water nexus of AMA. Moreover, there are many applications of DSM method (models and algorithms) that can create innovative insights. Adding new variables, such as energy coupling, financial coupling etc.; or elements, such as cleantech, business models, governance can unveil their impact on current Influence Profile, and can be compared.

5. Conclusions

The study allowed to unveil the complexity of the WW chain of AMA from an integrated perspective, and answered the research sub-questions with help of an MDM-framework.

The results show that physical connectivity, P concentration and ownership distribution play an important role in definition and organization of the system design and performance. A 100% P R&V target can be achieved by integrating and optimizing critical elements and dependencies in WW, E, drinking W, food and waste infrastructures, business models, products and services at household, neighborhood, city and regional levels. Finally, this study shows where to intervene first, which stakeholders to engage and how to leverage and optimize the current design for resilience and circular economy.

7. References

Amsterdam Institute for Advanced Metropolitan Solutions, 2016. Urban Pulse. Report.

Agudelo-Vera, C.M., 2012. Dynamic water resource management for achieving self-sufficiency of cities of tomorrow. Wageningen University and Research.

Eckert C., Clarkson P.J., Zanker W., 2004. Change and customisation in complex engineering domains. Research in Engineering Design 15 (1):1-21.

Eppinger S.D., Browning T.R., 2012. Design Structure Matrix Methods and Applications. Collection of publications about DSM method and application.

Henriquez L., Timmeren M., 2017. Under Pressure: Water and the City. TU Delft & AMS Institute.

Roefs I., Spiller M., Vreeburg J., 2016. Centralised, decentralised or hybrid sanitation systems? Economic evaluation under urban development uncertainty and phased expansion.

Schoumans O., Bouraoui F., Kabbe C., Oenema O., van Dijk, K., 2015. Phosphorus management in Europe in a changing world. *Ambio*, *44*(2), 180-192.

Spiller M., 2017. Measuring adaptive capacity of urban wastewater infrastructure – Change impact and change propagation. Science of the Environment.

Van der Hoek, J. P., Struker, A., & De Danschutter, J. E. M., 2017. Amsterdam as a sustainable European metropolis: integration of water, energy and material flows. Urban Water Journal, 14(1), 61-68.

Contact: M. V. Amosov, Organic Village, Pallieterstraat 25, 1183LA, Amstelveen, The Netherlands, +31 6 55 29 69 72, maxim.amosov@gmail.com

8. Appendix

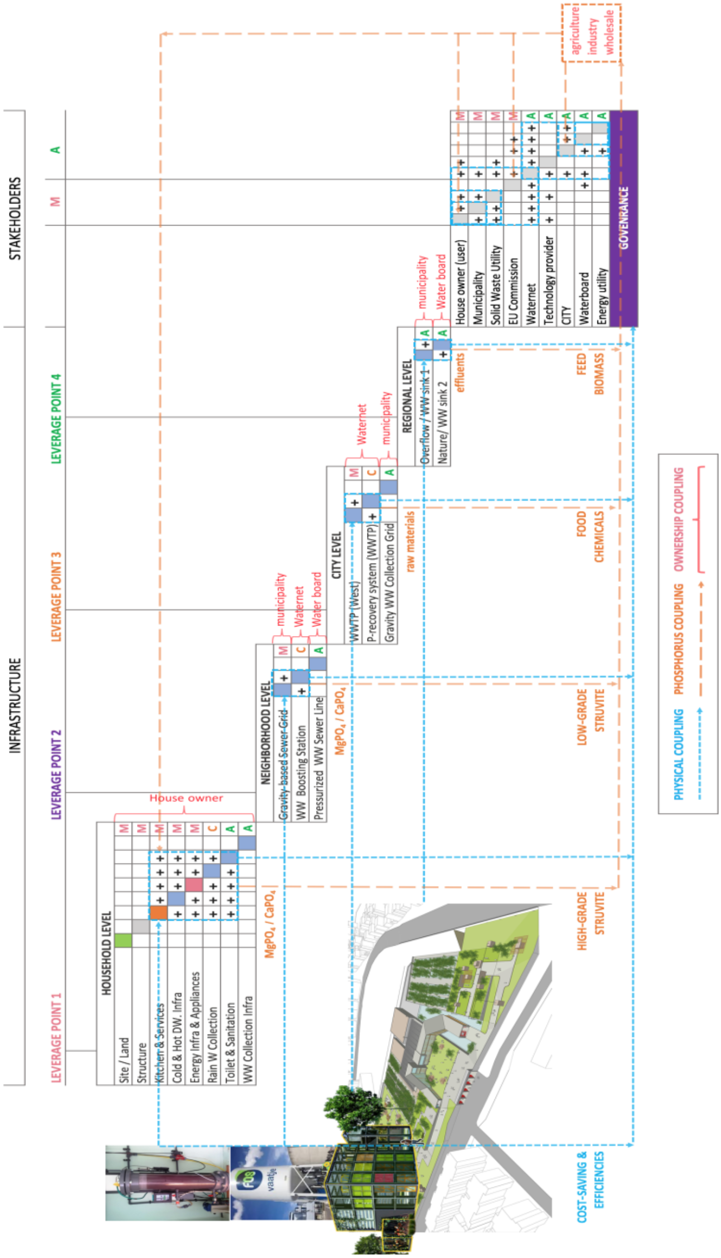


Figure 6. IP & a Roadmap towards 100% P regeneration at AMA (sample)

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Teaching DSM-based analysis based on a Case at a Start-Up

Marc Hirschka, Lucia Becerril

Laboratory for Product Development and Lightweight Design, Technical University of Munich, Germany

Abstract: Complexity Management Methods have been proven a useful methodology managing complex product architecture. However, insufficient training and lack of visibility of the advantages hinder the transfer and implementation of these methods. This paper presents a teaching case based on a real case that facilitates practical training on complexity management methods. The aim of the case is to develop an understanding of DSM and related methods among students and practitioners. The data was collected through the analysis of documents and a series of interviews within a start-up.

Keywords: Education, Case Study

1 Introduction

Design Structure Matrices or Dependency and Structure Modelling (DSM) has been proven a useful methodology for representing systems architecture (Eppinger and Browning, 2012) and managing complexity. For the past years, DSM and related methods have been applied in a broad range of industries. However, a widespread application in academia and industry is yet missing. According to Albers et al. (2013) and Becerril et al. (2017), insufficient training and lack of visibility of the advantages are major barriers to transfer and implement methods in industry.

The case method has been applied in many scientific fields as an effective learning tool since it combines what has been learned with real world problems and offers the learner sustainable learning through active involvement (Bonney, 2015; Popil, 2011). This type of knowledge transfer has been shown to increase the motivation of learners and achieve better learning success than conventional methods such as frontal instruction (Bonney, 2015; Popil, 2011). Thus, this paper presents a teaching case (TC) that facilitates hands-on training on complexity management methods. The aim of the case is to develop an understanding of DSM and related methods among students and practitioners, so that they are able to incorporate the methodology into their own projects. The TC includes DSMs, Domain Mapping Matrices (DMM) and Multiple Domain Matrices (MDM), as well as analytical methods. Moreover, during the assignment the applicability and usefulness of DSMs and related methods is demonstrated. In particular, cases offer the opportunity to demonstrate the connection between academic topics and the real world promoting the participants' understanding of the application (Bonney, 2015; Popil, 2011). Here, the participants should experience the advantages and limitations of the applied methods.

The TC reflects the results of a case study within a start-up and represents real requirements and conditions within the current development of a technical system. The data was collected through the analysis of documents and a series of interviews over the course of six months. Then, product architecture and organization's structure were modeled and analyzed.

2 Background

This chapter presents a brief introduction to DSM and related methods applied in the TC, as well as an overview on the topic of product architecture.

2.1 Product Architecture

For many years product architecture has been an important topic within product development, for instance as highlighted by Henderson and Clark (1990) and Ulrich (1995). According to Eppinger and Browning (2012, p. 7), system or product architecture (in this TC "product" and "system" are used as synonyms) is *"the structure of a system – embodied in its elements, their relationships to each other (and to the system´s environment), and the principles guiding its design and evolution – that gives rise to its functions and behaviors."*.

2.2 Design Structure Matrix (DSM)

The simplest form is the binary DSM (Yassine, 2004, p. 2). A relation between two elements is marked by "x" or "1" in the respective cell (Warfield, 1973). However, further representations are highlighted for example by Eppinger and Browning (2012, p. 5) and Clarkson et al. (2004). The DSM can help to increase understanding of the system, which results from the complex relationships between individual system components. The product architecture DSM depicts the relationships between the components of a complex system or product.

Domain Mapping Matrix (DMM)

Danilovic and Browning (2004) introduced the DMM as a complementary approach, where connections between two different domains can be visualized, e.g. components and people responsible for those components. The DMM is established as a helpful extension of DSMs in times of increasingly complex structures (Browning, 2016, p. 27). The DMM is formed by relating two different DSMs. The resulting DMM is usually rectangular (m x n), where m is the size of the first DSM and n is the size of the second DSM (Danilovic and Browning, 2007, p. 302). The analysis options for DMMs are not extensively addressed in literature so far.

Multiple Domain Matrix (MDM)

The MDM combines DSMs and DMMs in one representation. The DSMs are arranged in diagonal and the DMMs are arranged off-diagonal. In other words, it represents a DSM at a higher level of abstraction, namely between domains and not, as before, elements within a domain. Thus, a holistic overview of product development is created.

All these matrices capture, display, process, and analyze complex systems. Alone through identifying the relations and presenting the system's structure, the matrices increase system understanding and facilitate communication with others stakeholders. Analyzes allow even more in-depth knowledge about the system under consideration. (Yassine, 2004, p. 15).

2.3 Analysis Methods

Here, three analysis methods that are applied in the TC are briefly described. An overview of further analytical methods is provided in Browning (2016, p. 29).

Clustering

The most common analysis method, especially for product architecture analysis, is the clustering method (Browning, 2016, p. 30). *"Clusters represent a basis for creating modules."* (Lindemann et al., 2009, p. 227). Hence, it helps to find modules of subsystems or components which are closely linked among them and slightly linked to further modules (Yassine, 2010, p. 319). Thus, a cluster combines one or more components whereas *to cluster* means the creation of *"... a set of Clusters by means of an algorithm."* (Börjesson Frederik, 2012, p. 3). Further information is given in Sharman and Yassine (2004).

Influence Portfolio

The influence portfolio analysis enables a clear graphical representation of the components on the basis of their influence and their influenceability (Probst and Gomez, 1991, p. 14). For this purpose, the active sum and the passive sum of each component is formed (Melnikov et al., 1994, p. 279). After visualizing the DSM each row and column is summarized. In case of the row total of a component the term active sum is used. Accordingly, the term passive sum is used in case of the column total (Probst and Gomez, 1991, p. 189). These values are then entered in the influence portfolio. The x-axis corresponds to the active sum and the y-axis to the passive sum. To classify the components it is helpful to divide the influence portfolio into different areas. (Lindemann et al., 2009, p. 162).

Indirect Relations

The indirect relations analysis helps to uncover relations that exist indirectly between two components (Lindemann et al., 2009, p. 99). Especially the indirect dependencies with one intermediate component are of interest. To apply indirect relations analysis, the output matrices must be prepared correctly through data acquisition. This means that only direct relationships between components should be depicted. Hereafter, the deduction of indirect relations must take place. Effectively, a matrix multiplication is carried out. (Eichinger et al., 2006, p. 232).

3 Methodology

This section provides an overview on the methodology behind this paper. The process comprised following three phases: data collection, modelling and model verification, and TC development.

3.1 Data Collection and Modeling

Data collection, modelling and model verification was an iterative procedure. The data was collected through reviewing existing documents, inspecting physical products, and a series of interviews with domain experts. In this case, the main advantage of the interviews was that each expert could be addressed individually. In addition, questions from the expert on the method or the interviewer on detailed information or unclear relations could be answered immediately. These advantages are also mentioned by Eppinger and Browning (2012, p. 40) and Moon et al. (2015, p. 328). The interviews were conducted based on the approach by Moon et al. (2015, p. 327), whereby the following two adjustments were made. First, instead of preparing and conducting a questionnaire an interview is prepared and performed. Second, instead of conducting a consensus round survey to elucidate identified items, these items were clarified by direct discussions with the corresponding experts.

After initially brief discussions with the respective experts to obtain an overview of the system, documents were sighted, and the physical product was examined. After this phase the product could be divided into individual hardware (HW) and SW functional units (SW-FUs). Also, some spatial relationships could already be identified and documented in a first version of the spatial DSM. From the discussions and documents, preliminary information about the information flow could also be collected and illustrated in the corresponding software DSM "information flow". In this early phase, however, the main goal was the formation of the MDM, which originally was created based on the collected information and literature research on 13 different domains. Here, the domains HW components, SW components and persons were selected to create the TC, and for further analysis within the company.

Afterwards, the relations for the connections between the selected domains were determined. In the HW and SW domains, the relations "geometrical constraints" and "information flow" were considered. In relation to the domain "Person", the relations to be considered were "communication" (DSM) and the responsibilities towards HW and SW components (DMMs).

After the MDM and the first DSMs and DMMs were created, five domain experts were interviewed for ca. 1.5 hours each. The experts completed and verified the previously acquired HW DSM (consisting of 50 components), SW DSM (19 components), and HW-SW DMM matrices. Furthermore, during the interviews most relations among components were verified. Connections between components that could not be verified were then directly discussed with the corresponding domain experts. For all matrices created during the case study we use the binary representation and the IC convention, where the input is mapped in the columns and the output in the rows.

3.2 TC Development

The target group of the teaching case are primarily mechanical, mechatronic and electric engineers or engineering graduate students. The main requirements are an understanding of the architecture of mechatronic systems and basic knowledge of DSM and related methods, which in our case is given in a previous lecture. . Participants should experience how DSM methodology can be applied in their projects. Overall, the DSM methodology can provide a deeper and structured understanding of the interdependencies between components independently of the level of detail.

The TC was developed according to the framework by Kim et al. (2006). This framework describes the basic division of different strategies for TCs into four categories. Figure 1 shows an overview of the framework. Categories one (content), two (structure) and four (process) comprise 17 strategies that ensuring that the five core attributes (third category) are met (c.f. Kim et al., 2006, p. 869). In our contribution, the strategies highlighted in Figure 1 were applied to create a successful TC.

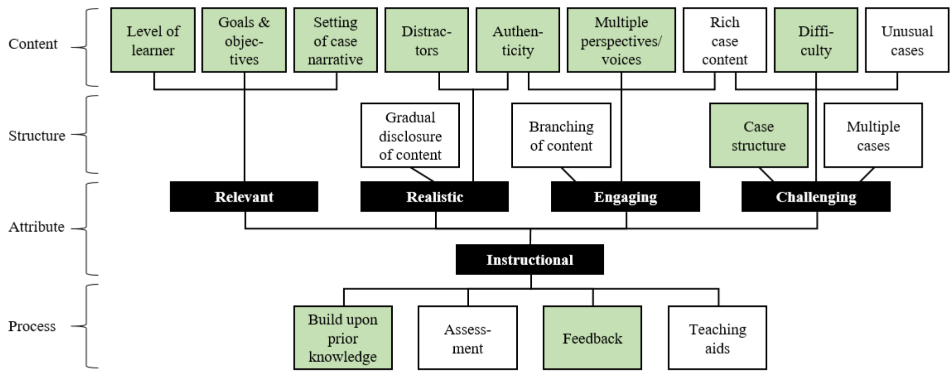


Figure 1: Conceptual framework of TC development based on Kim et al. (2006, p. 869).

The TC has been applied in three different workshops, two of them with ca. 15 mechanical engineering undergraduate and masters students each and a third occasion with ca. 20 PhD Students from the field of Systems Engineering. Thus, its applicability has been evaluated. Slight changes were made to the original TC according to the participants' feedback.

4 Teaching Case

Based on the expertise which was collected during the work within the start-up the following TC was built. The case is planned for a 3-hour workshop and the participants shall have basic knowledge on complexity management and product architecture (in the workshops we conducted this was achieved by a 1.5 hours lecture beforehand for students with engineering background.). Due to confidentiality the terms for modules and components and the names of persons have been modified.

4.1 Introduction

The start-up develops complex technical products with interconnected HW components and SW-FUs. The product is a parking spot sensor system which collects and evaluates data in real-time, thus, knowing the availability of appropriately equipped parking spots. This information can then be used, for example, by a navigation device to navigate the car driver directly towards a free parking spot. To provide this possibility, the system consists of a base STATION unit and SENSOR unit. For both HW components a SW is mandatory for a functional system. As a third component, we also consider the HW and SW employees in their respective functions in the TC.

In Figure 2 the system boundary of the considered system is pictured. The influence is displayed across the system boundaries by three different arrow types and the corresponding arrow direction. The considered domains and their relationship to each other are displayed in the MDM (Figure 3). The color of the relationship indicates whether the required DSMs and DMMs already exists (black) or must be compiled independently during the TC from additional information (red).

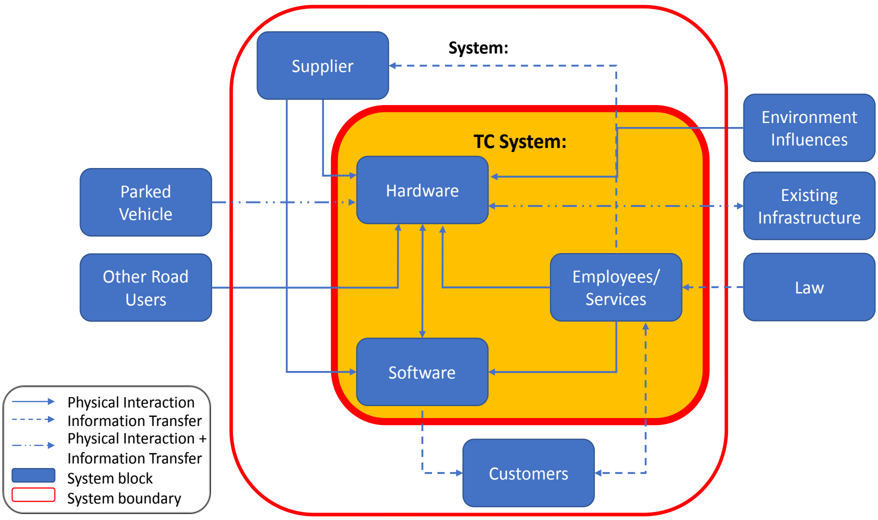


Figure 2. System boundary, the considered system (TC System) consist of HW and SW-FUs as well as employees

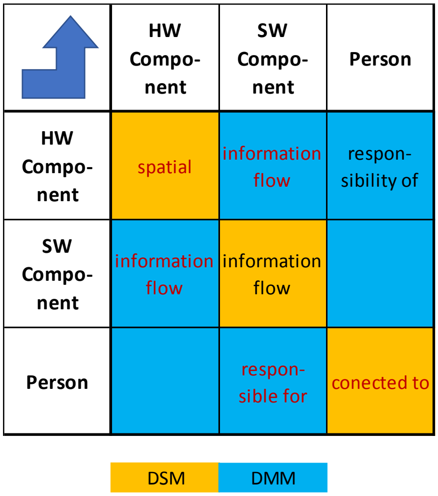


Figure 3. MDM TC with prefilled relations (black), to complete relations (red)

4.2 Preparation

A brief introduction of the situation, environment, and the expected performance as well as all necessary information to conduct the TC is presented in seven worksheets (WSs). Table 1. Contents of the individual worksheets presents the contents of the individual worksheets.

Furthermore, practical solutions are developed for three problems, which are described below. Notes on the WSs as well as practical solution sheets are intended for clarification purposes and assistance for participants and teachers. The seven WS and the three solution proposals can be obtained by email to the authors if required. In addition, a spreadsheet program is required for the analyses and best case, the program is accessible to every participant at the same time.

Table 1. Contents of the individual worksheets

WS	Content
0	Introduction text (scenario); Issues (1-3); MDM (completed for TC)
1	eMail of the Sensor-Expert to complete the HW DSM "spatial" on WS2
2	HW DSM "spatial" to be completed with information of WS1
3	SW DSM "information flow" (completed)
4	Network diagram of employees related to SW-FUs; DMM HW-Person "responsibility of" (completed)
5	Network diagram of the information flow through SENSOR and base STATION units
6	Person DSMs "connected to" to be completed

The first task is to find HW modules through clustering of the HW DSM “spatial”. Therefore, the HW DSM “spatial” must be partially self-developed.

The second objective is to identify and display the critical, active, passive and inert (Lindemann et al., 2009, p. 162) components within the SW DSM “information flow”. Therefore, the active sum and passive sum of each component of the SW DSM “information flow” must be calculated and displayed in a portfolio. In addition, the threshold must be set for active, passive and critical components. Furthermore, the criticality of each component can be calculated and used to scale the size of the component within the influence portfolio.

The objective of the last task is to find dependencies between elements through indirect dependency analysis. Hence, the correct matrices must be identified, multiplied with each other, and insights must be drawn.

A summarized representation of the issues together with the questions dealt with as well as the related analysis methods and the related WSs is given in Table 2.

Table 2. Considered issues, discussed questions, analysis method, and related worksheets

Issue	Question	Analysis Method	Worksheets
1	Which HW components can be clustered?	Clustering	WS1; WS2
2	Which are the inert, passive, active, and critical components?	Influence Portfolio	WS 3
3	Which persons are connected regarding their component responsibilities?	Indirect Dependencies	WS2; WS3; WS4; WS5; WS6

4.3 Conducting the workshop

The participants are ask to envision themselves working in a tech start-up which develops parking spot sensors. They are asked to carry out a product architecture analysis of the sensor system to tackle some ongoing challenges in the start-up. For example, they should give a recommendation on who should be responsible for which HW modules. At the beginning of the workshop, the MDM in Figure 3 is given as well as all worksheets in Table 1 (WS1 – WS6). They are recommended to use a spreadsheet program and work in teams of 3-5 people.

Participants should have a basic understanding and/or receive a brief introduction to the DSM subject. In the TC, the focus lies in modelling a technical system using matrix based approaches and on applying the methods clustering, analysis of indirect dependencies and portfolio analysis (c.f. section 2). In the overview WS (WS0), the scenario and MDM (Figure 3) as well as the problems to solve are presented and the participants can work independently. However, for the instructor of the TC it might be useful to interrupt the TC between the tasks, e.g. to compare the groups result, answer questions and proceed with the same base. At the end of the workshop the questions of the participants shall be answered, the results compared and discussed. Thus, the participants shall reflect on the methods applied and their results – with the objective of increasing the understanding of the applicability of the methods and supporting the learning process.

4.4 Learnings in conducted workshops

Incorporated later, the first excersise comprises filling out a part of the DSM. This helped participants become familiar with the technical system and the modelling approach. Giving an incomplete DSM at the beginning helped balance the time necessary for the exercise with gaining understanding about the system to analyse. Furthermore using tools the participants are already familiar with, such as spreadsheets, allows the participants to focus on the excersice rather than on the tool. Depending on the level of experience of the participants, we gave hints on which analysis method to use for every question. However, experienced participants would have benefited from it as well.

5 Conclusion and Outlook

The goal of this paper is to present a realistic TC for selected complexity management methods. For this purpose, the results and experience gained during the application DSMs, DMM, MDMs and analytic methods within a start-up were used to develop a realistic and engaging TC. Based on the TC, the participants can apply these methods on a practical example gaining a deeper understanding. Moreover, the experienced insights from analysis ideally motivate the participants to apply the methodology in their daily work. To gain these benefits, the participants should receive an introduction on the topic and the basic procedures beforehand, including the analysis methods such as clustering analysis, influence portfolio analysis, and indirect relations analysis. Within the TC the topics of product architecture analysis and employee-component assignment are addressed, as well as derived relations among employees. Further applications of DSM-related methods are not included. Moreover, the three analysis methods clustering analysis, influence portfolio analysis, and indirect relations analysis are applied.

The practicality of the TC is ensured through a number of strategies as shown in Figure 1. Additionally, the TC has been evaluated in three practical applications and continuously improved. Further applications will include and a follow-up questionnaire to evaluate the TC according to the five attributes by Kim et al. (2006) and its long-term usefulness.

Acknowledgements

The authors would like to thank Sebastian Schweigert and Sjoerd Knippenberg for their support conducting part of the workshops.

References

Albers, Albert, Lohmeyer, Quentin, and Radimersky, Ahne. (2013): "Individuelle und organisatorische Akzeptanz von Metho-den des Systems Engineering." Tag des Systems Engineering 2013: Zusammenhänge erkennen und gestalten 351.

Becerril, L., Hollauer, C.,Kattner, N., Weidmann, D., Lindemann, U. (2017) How to assess the acceptability of design support. Tag des Systems Engineering 2017. Hanser.

Bonney, K. M. (2015). Case study teaching method improves student performance and perceptions of learning gains. *Journal of microbiology & biology education*, *16*(1), 21–28. https://doi.org/10.1128/jmbe.v16i1.846

Börjesson Frederik. (2012). Approaches to modularity in product architecture. TRITA-MMK: 2012:11. Stockholm: Department of Machine Design, Royal Institute of Technology. Retrieved from https://modularmanagement.com

Browning, T. R. (2016). Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. *IEEE Transactions on Engineering Management*, *63*(1), 27–52. https://doi.org/10.1109/TEM.2015.2491283

Clarkson, P. J., Simons, C., & Eckert, C. (2004). Predicting Change Propagation in Complex Design. Journal of Mechanical Design, 126(5), 788–797. https://doi.org/10.1115/1.1765117

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

Danilovic, M., & Browning, T. R. (2007). Managing complex product development projects with design structure matrices and domain mapping matrices. *International Journal of Project Management*, *25*(3), 300–314. https://doi.org/10.1016/j.ijproman.2006.11.003

Eichinger, M., Maurer, M., & Lindemann, U. (2006). Using multiple design structure matrices. Advance online publication. https://doi.org/10.1002/0471776688.ch1

Eppinger, S. D., & Browning, T. R. (2012). *Design structure matrix methods and applications*. *Engineering systems*. Cambridge, Mass.: MIT Press. Retrieved from http://lib.myilibrary.com/detail.asp?id=365529

Henderson, R. M., & Clark, K. B. (1990). Architectural Innovation: The Reconfiguration Of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly*. (35), 9–30.

Lindemann, U., Maurer, M., & Braun, T. (2009). *Structural Complexity Management: An Approach for the Field of Product Design*. Berlin, Heidelberg: Springer. http://dx.doi.org/10.1007/978-3-540-87889-6

Kim, S., Phillips, W. R., Pinsky, L., Brock, D., Phillips, K., & Keary, J. (2006). A conceptual framework for developing teaching cases: A review and synthesis of the literature across disciplines. *Medical Education*, *40* (9), 867–876. https://doi.org/10.1111/j.1365-2929.2006.02544.x

Melnikov, O. I., Tyshkevich, R., Yemelichev, V., & Sarvanov, V. (1994). *Lectures on graph theory*. Mannheim: Wissenschaftsverlag.

Moon, J., Lee, D., Lee, T., Ahn, J., Shin, J., Yoon, K., & Choi, D. (2015). Group Decision Procedure to Model the Dependency Structure of Complex Systems: Framework and Case Study for Critical Infrastructures. *Systems Engineering*, *18*(4), 323–338. https://doi.org/10.1002/sys.21306

Popil, I. (2011). Promotion of critical thinking by using case studies as teaching method. *Nurse education today*, *31*(2), 204–207. https://doi.org/10.1016/j.nedt.2010.06.002

Probst, G. J. B., & Gomez, P. (1991). Die Methodik des vernetzten Denkens zur Lösung komplexer Probleme. In G. J. B. Probst & P. Gomez (Eds.), *Vernetztes Denken: Ganzheitliches Führen in der Praxis* (2nd ed., pp. 4–20). Wiesbaden: Gabler.

Sharman, D. M., & Yassine, A. A. (2004). Characterizing complex product architectures. *Systems Engineering*, *7*(1), 35–60. https://doi.org/10.1002/sys.10056

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, *24*(3), 419–440. https://doi.org/10.1016/0048-7333(94)00775-3

Warfield, J. N. (1973). Binary Matrices in System Modeling. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*(5), 441–449. https://doi.org/10.1109/TSMC.1973.4309270

Yassine, A. A. (2004). An Introduction to Modeling and Analyzing Complex Product Development Processes Using the Design Structure Matrix (DSM) Method. Retrieved from https://www.researchgate.net

Yassine, A. A. (2010). Multi-Domain DSM: Simultaneous optimization of product, process and people DSMs. In Managing Complexity by Modelling Dependencies: Proceedings of the 12th International DSM Conference (pp. 319–332). München: Carl-Hanser-Verlag. Retrieved from https://www.designsociety.org

Contact: Lucia Becerril, Technical University of Munich, Laboratory of Product Development and Lightweight Design, Boltzmannstr. 15, Garching bei München, Germany, lucia.becerril@tum.de, mw.tum.de/en/lpl/home

20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018
TRIESTE, ITALY, OCTOBER 15 – 17, 2018

Model-Based Consistency for Design for Variety and Modularization

Michael Hanna, Lea-Nadine Schwede, Dieter Krause

Hamburg University of Technology

Abstract: Methods for developing modular product families contain a lot of information and data that are not always consistent. Consistent data modeling enables simple changes across all affected tools while developing modular product families. Redundant information can be identified and eliminated, and networking between different data is enabled. Inconsistency in variant management is analyzed and addressed with a data model that supports consistency at different hierarchical levels. The data model is modeled with the Cameo Systems Modeler in SysML and is used in a case study for a Design for Variety and modularization with DSM on a vacuum cleaner robot.

Keywords: Model-based approach, Design for Variety, Modularization, Consistency, Data model

1 Introduction

Methods for developing modular product families are characterized by the use of a large amount of information and data, which can include numerous components, their links, and the resulting modules. Product development processes such as variant management (for example, in the method Design for Variety) are often inconsistent because they are carried out in a document-centered manner in companies. Methodological tools do not relate to each other or to available company information. A way to create and document different stages of development is also not provided. Inconsistencies can occur and networked information cannot be understood, which hinders the development process of modular product families. Consistency is important to ensure changes throughout all relevant documents and networking between the pieces of information, as well as to avoid redundant information. (Albers and Lohmeyer 2012, Krause et al 2014, Bursac 2016)

This paper addresses the problem using a model-based approach in which the data is modeled in a consistent data model to counteract inconsistencies that have occurred in product development to date. Variant management and the development of modular product families are modeled on a Design Structure Matrix (DSM). Tools from Model-Based Systems Engineering (MBSE) are used, because they aim for better document management.

Methods for variant management and the current state of research in MBSE are explained. Then the consistent data model and its implementation in Cameo Systems Modeler are shown. The resulting model is used to reduce internal variety in the example of vacuum cleaner robots.

2 Methods of Variant Management and Terms of MBSE

This section presents methods for reducing internal variety and some relevant terms of MBSE.

2.1 Methods for reducing internal variety

High external variety results from a multitude of variant products being offered by a company and leads to high internal variety. Variants could be managed using the method unit Design for Variety from the Integrated PKT-Approach. Its aim is to change the product structure to achieve the ideal of a variety-oriented product structure. This method is supported by some visualization tools (Figure 1).

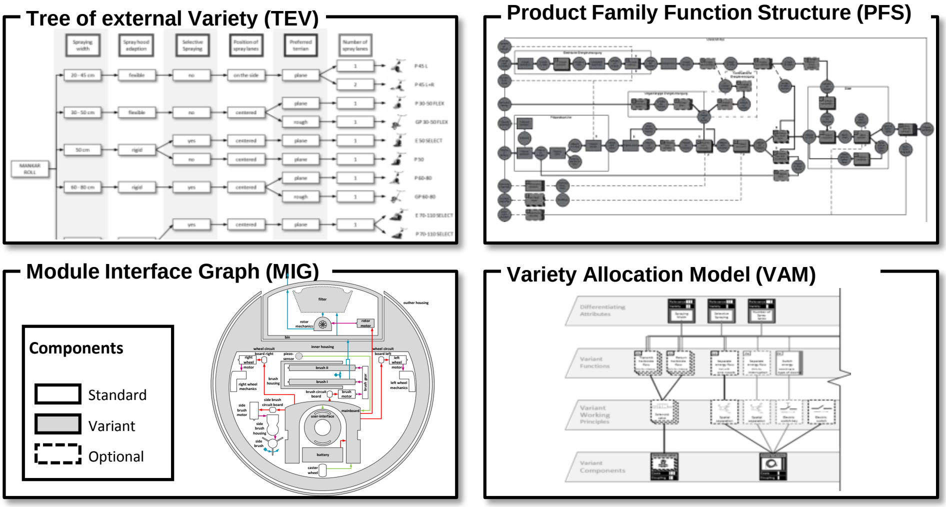


Figure 1. Tools for Design for Variety, according to Gebhardt et al 2012

External variety is represented by the Tree of External Variety (TEV); internal variety of components is shown in the Module Interface Graph (MIG) and the internal variety of functions is viewed in the Product Family Function Structure (PFS). In the Variety Allocation Model (VAM), elements of these tools are linked to each other. TEV, MIG, PFS and VAM are partial models. (Krause et al 2014)

Another way to reduce internal variety is by using a modular product structure. Modularization can be carried out based on Design for Variety. There are different methods for modularization; three are presented below.

Modularization based on the needs of all life phases can be developed with the method unit Life Phases Modularization of the Integrated PKT-Approach (Krause et al 2014). With the procedure of Stone, module formation is based on the flows within the functional structure. Three heuristics are determined: dominant flow, branching flow and conversion-transmission (Stone 1997). The basic premise of the Integration Analysis methodology, based on the Design Structure Matrices (Steward 1981), is that components exhibit the technical couplings information transfer, energy transfer, material transfer and spatial dependencies. The stronger that components are coupled the more they should be part of the same module (Pimmler and Eppinger, 1994).

2.2 Terms of Model-Based Systems Engineering

A model is an abstraction of reality which has three main characteristics. Representation means that models represent something; Reduction indicates that not all attributes are represented in the model, only the relevant ones; Pragmatism means that models are not clearly assigned to their originals. A meta model is itself a model that can be used to describe modeling. Here, the term data model is used, which is at an abstract level and is understood as a meta model. (Holt et al. 2012, Stachowiak 1973)

Consistency of time means that models can be used across different times; they are extendable. Consistency of vertical levels means that the model has to be vertically consistent. Consistency between different models of one product family is required (Bursac 2016, Scherer 2016). In their consistency management, Herzig et al. focus on the early identification of existing inconsistencies. Levels of inconsistency can be classified as internal or external inconsistency (Herzig et al 2011).

Systems engineering supports the developing process of systems, according to systems requirements. It consists of system architecture, system requirements and system behavior. With MBSE, system elements can be modeled and information can be linked, so that information can be saved and used in a networked model. The language SysML was developed for MBSE. The modeling software Cameo Systems Modeler uses SysML notation as well as diagrams and tables. (Weilkins 2008, Holt et al. 2012)

Previous work using model-based approaches in modular product development concentrated on either variant management (see Bahns et al 2015, Hanna and Krause 2017) or developing modular product families (Bursac 2016, Scherer 2016).

3 A Data Model for Consistent Development of Modular Product Families

The methodical approach is shown in this section. The challenges of data inconsistency, the procedure for developing a data model and the advantages of the data model are shown.

3.1 Challenges of Data Inconsistency in Product Development

A data model that supports consistency is needed. For example, Design for Variety in the Integrated PKT-Approach consists of several inconsistent partial models that are used mainly in Microsoft Powerpoint printouts. For example, the variant product properties that are created in the TEV are inserted at the top level of VAM. This does not happen automatically, which can lead to errors and high overhead during transmission. The same applies to the functions that are shown in the VAM at the middle level and come from the PFS. The components in the VAM are not consistently extracted from the MIG. In case of later changes, such as a change of component, all contained tools must be modified (in this case, VAM and MIG). Creating new versions is expensive and can lead to errors. There is no link between variant management and modular product family development. Beyond this, in the different tools the versioning is only partial available, but not consistent.

3.2 Procedure for Modelling a consistent Data Model

The following steps have been taken to shift from inconsistent partial models to a consistent data model for variant management. First, all used methods (for example, Design for Variety and Integration Analysis Methodology) and their tools (for example, the VAM and the DSM) are analyzed. Elements and how they link to each other are identified. Based on this, the data model was then built based on the components, because the components are found in many tools and form the center of the methods. The links were not defined exactly in the data model in order to remain as solution-neutral as possible in the modeling software implementation..

3.3 Data Model

A data model is needed to solve the consistency problems described in Section 3.1. The three levels of consistency mentioned above were taken into account: Consistency of time, vertical levels and between different models. The focus is not on the rapid identification of inconsistencies, as in Herzig et al., but on building a new consistency model, as one does not yet exist (Herzig et al 2011). The data model for both Design for Variety and modularization is shown in Figure 2.

The top of Figure 2 shows the elements and models needed for the method Design for Variety; the bottom shows the ones for modularization. Every element only exists once, in contrast to previous unlinked tools. Design for Variety and modularization can be considered as sub-models of the data model and include further models and elements. At their overlap area, components are presented once; they form the central elements here. In Design for Variety they are used in the MIG (III, colored green), where they are connected to the flows and in the VAM (IV, colored blue), where they are linked with the work principles. In the modularization the components are used in the DSM (V, colored red), where they are linked to the couplings and the modules. The tools used here are partial models because they are an abstraction of reality (for example, the MIG is an abstraction of the real product family).

All elements used are shown and linked to each other. Each element is built with four layers for the different versions. Elements can have different versions: before and after Design for Variety; and after modules are built and changes made. Each layer represents one of these versions. If an element exists in several versions, then the corresponding layers are colored gray.

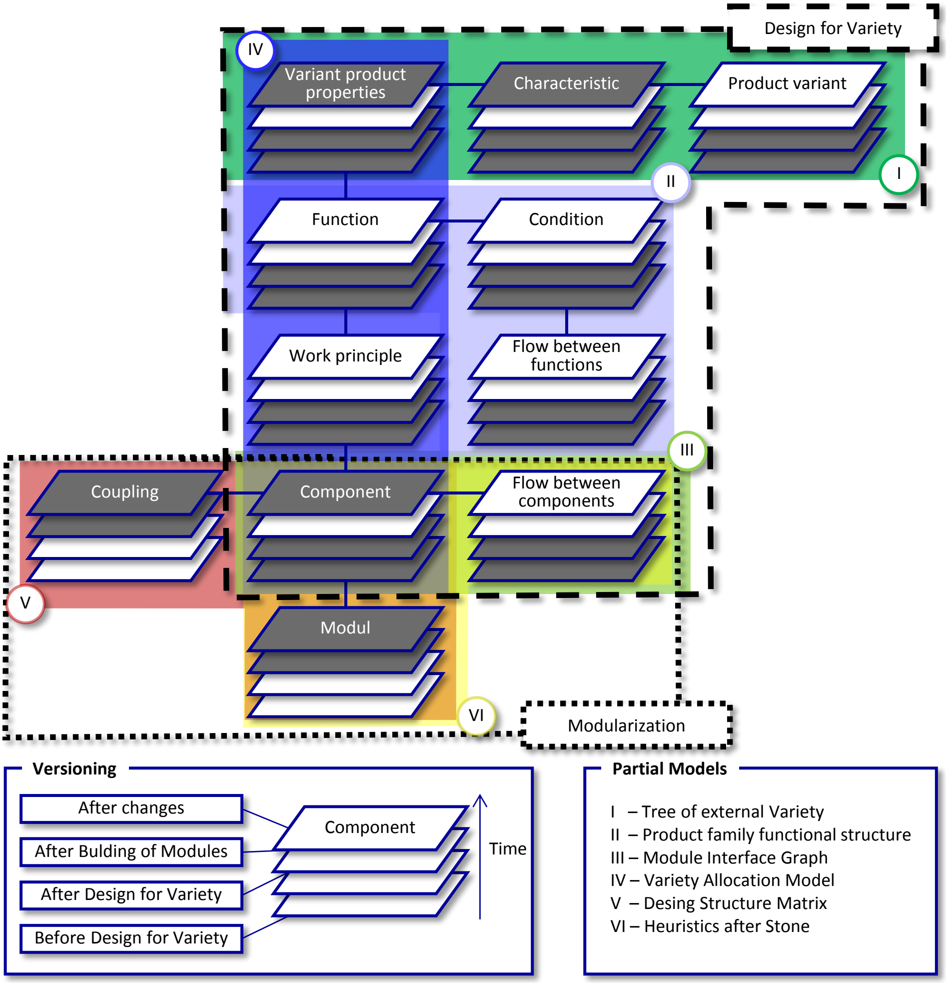


Figure 2. Consistent data model for Design for Variety and modularization

Components in Design for Variety may change. The type of variance can be changed constructively to achieve a variant-oriented product structure. For example, a component can be available in several variants at the beginning and standardized after Design for Variety. This is why the two lowest layers of the model element “component” are marked in gray in Figure 2. There may be one version before and another after using the method. After building modules, the components do not change with the modularization methods used. The third level is marked white. If there are subsequent changes, components can be changed again, which is why the top layer is highlighted in gray. In contrast to components, modules are not used during variant-oriented product design, which is why the two lowest layers are white. In addition, there is a version after modules are built, so the third layer is colored gray. After changes in the modular product structure, modules can be changed, so the fourth layer is also gray.

3.4. Advantages to the consistent data model

For the first time, a consistent data model for variant management and modularization has been created in which all the elements used are consistently linked. Elements that are used in both areas can be easily identified (for example, the components). This data model is an expandable meta model for the product development of modular product families. Based on the data model, consistent modeling of Design for Variety and modularization can now be achieved using MBSE. The data model was implemented in Cameo Systems Modeler.

The three levels of consistency are achieved as follows:

- Consistency between models: Because every element exists only once, changing an element in one partial model definitely leads to the same element changing in another partial model. Once an element (e.g. component) has been created correctly, it is correctly created in all partial models (e.g. MIG, VAM and DSM).
- Consistency of time: Using different layers, the different versions in which an element can appear are figured out. Which elements change over time becomes clear. Components change in Design for Variety and after changes.
- Consistency of vertical layers: Vertical consistency can be achieved using different hierarchical levels of the models. For example, the partial model DSM and the Heuristics of Stone are both part of modularization, which is part of the whole data model; they all use the same component.

4 Case Study based on the Data Model, using SysML

This section shows application of the data model using the Cameo Systems Modeler. A simple case study on the vacuum cleaner robot demonstrates application of the individual tools of Design for Variety and modularization comprehensibly. To generate a matching SysML model in Cameo Systems Modeler, first the tools are analyzed for contained elements and links between them. This produces the requirements for the SysML diagrams to be used.

For example, in the TEV allocations between the elements variant product properties, characteristics and product variants are used. This can be managed with block definition diagrams. An internal block diagram is used for the MIG because it can show different types of flows between the elements (in this case, components). In addition, the VAM was modeled. Besides the Design for Variety tools, modularization tools can also be modeled. In this case, the DSM, with its couplings between components, is realized with a dependency matrix in the Cameo Systems Modeler.

Individual partial models are implemented by different diagram types. They all use centrally stored elements that are linked to each other. Different versions of the individual elements are stored in separate packages in the model to allow versioning.

4.1 Modeling Design for Variety tools

The first step in Design for Variety is analyzing external variety using TEV. The MIG is used to analyze internal variety. Figure 3 shows implementation of the vacuum cleaner robot MIG in SysML. A detailed section is shown to clarify connections.

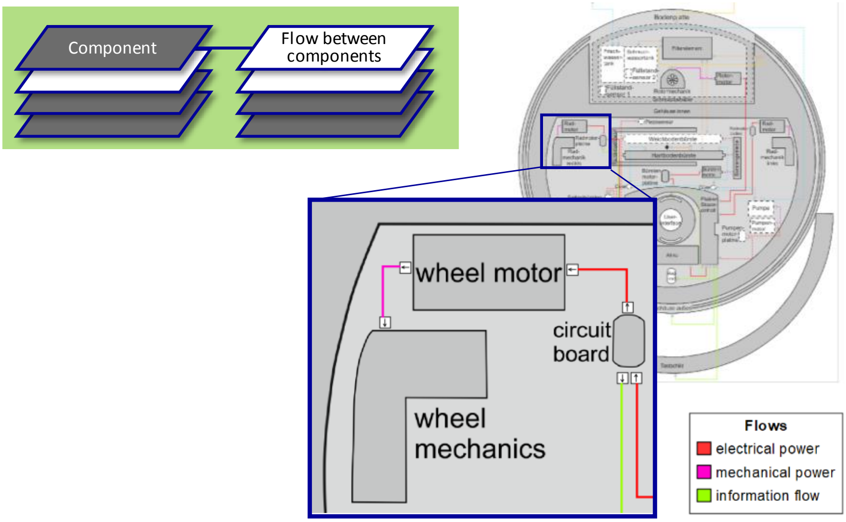


Figure 3. Module Interface Graph for the vacuum cleaner robot, modeled in SysML

The MIG for analyzing internal variety includes components that are linked to each other via flows. Pictures have been added to the individual elements and colored flows are linked to the components via ports. The components get their shape from the existing physical product and are shown in the color sheme, with gray colored components for variant components like “wheel mechanics” or “wheel motor”.

Together with a functional structure, these tools lead to the VAM (Figure 4), where the model elements variant product properties from the TEV are linked to the functions from the PFS. The functions, in turn, are linked to the work principles, which are linked to the components of the MIG.

The connections between the elements in the VAM are arranged vertically. The variant product property “Remote control” is linked to the function “Remote Control Detection”, and the function “Infrared Signal” is linked to the component “Infrared Receiver”. The variant product properties are characterized using a colored icon, which is also used for the connections between levels. For work principles, functions and components, the type of variance is pictured with a small icon in the upper right side. For example, the “Infrared Receiver” has a white symbol with dashed edge, which indicates that it is an optional component. The gray icon in the upper right corner of the component “Battery” indicates that it is a variant component.

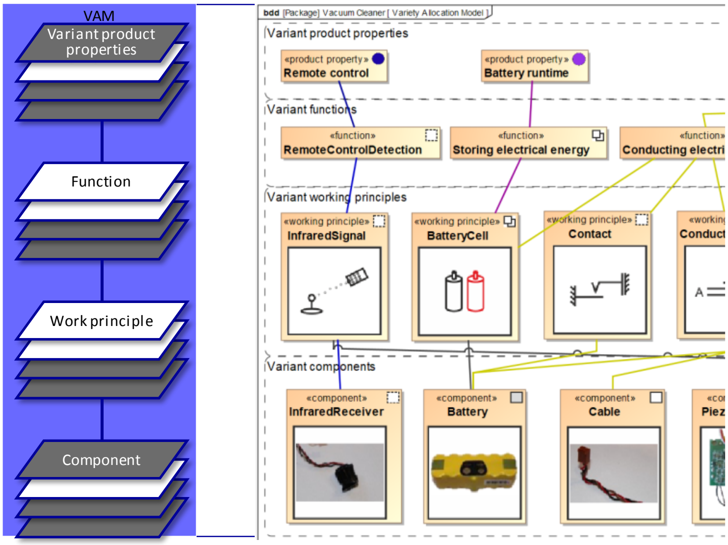


Figure 4. Variety Allocation Model implemented in SysML

4.2 Modeling Modularization with the Design Structure Matrix

Various methods can be used for modularization of a product structure. A DSM for modularization of the vacuum cleaner robot is shown here. The DSM includes components, couplings and modules. Figure 5 shows how these elements connect to each other.

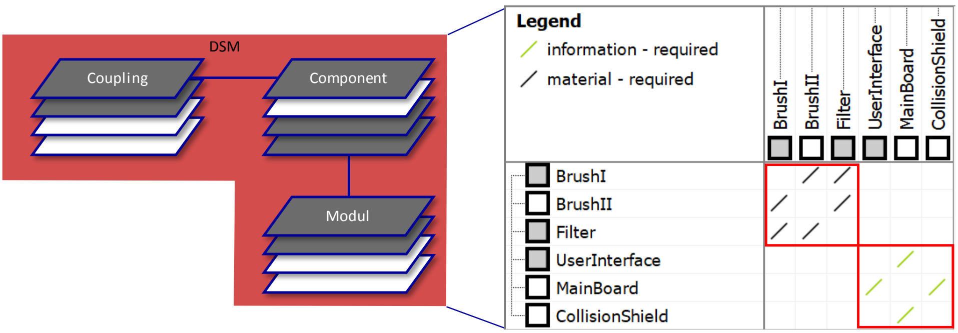


Figure 5. Detail of a Design Structure Matrix for a vacuum cleaner robot, modeled in SysML

Figure 5 is a detail from the DSM in Cameo Systems Modeler, showing the material and information coupling between components of the vacuum cleaner robot. The components “BrushI”, “BrushII” and “Filter” have a required coupling for material transport, and can be built to modules. The components “User Interface”, “Main Board” and “Collision Shield” are linked through required information flow.

In addition to the standard DSM, the type of variety is shown next to the components. For example, next to the component “BrushII” a white box is shown, which indicates that this is a standard component. A gray box next to component “BrushI” identifies this component as variant.

4.3. Evaluation of the application in SysML

By developing a data model the data can be managed. A data model makes it clear that components need only be present once, but can be used in different tools. A new level of continuity can be achieved. If a component is to be changed, the change is visible in every diagram where the component is used. However, visualization of individual tools is no better than conventional visualization, such as in PowerPoint. An advantage of using Cameo Systems Modeler is the possibility of adding additional information to individual model elements and defining connections between model elements that are consistent throughout the whole model. For example, information on variance of a component can now also be shown in a DSM. This makes the DSM more meaningful and supports module building. Information, which is generated in the DSM can be easily added to the model itself.

5 Conclusion

Design for Variety and modularization can be supported with a consistent data model. Methods used previously to reduce internal variety led to inconsistencies in the creation and modification of elements. Three levels of consistency are made possible with a data model. Model elements were linked consistently to Design for Variety and integration analysis methodology tools. Modeling of Design for Variety and modularization in one MBSE model, using the Cameo Systems Modeler, was presented. The SysML model was used in the case study on a vacuum cleaner robot and showed how the Design for Variety and modularization tools can be visualized and made consistent.

The data model can now be changed at a central location, which results in automatic change of the partial models used. Various versions can now be displayed in one model. In addition, DSM work is supported as not only individual products but also entire product families can be modularized, as the type of component variance becomes clear when combined with variant management.

The data model needs to be expanded with further research. By including life phases, the Life Phases Modularization can be added and more methods for developing modular product families used. Cost information can be added to the data model and through life phases, possibly via the Impact Model (Hackl and Krause 2017). Inclusion of the Impact Model in the data model is feasible, as it access the (modular) product structure. It also contains a lot of information, which could be handled with a continuous sub-model. The connection of boundary conditions is also relevant. Since the Impact Model contents are partly based on mathematical key figures, a link to Matlab is conceivable.

References

Albers, A., Lohmeyer, Q., 2012. Advanced Systems Engineering – Towards A Model-Based and Human-Centered Methodology. Proceedings of the Symposium on Tools and Methods of Competitive Engineering (TMCE), Karlsruhe.

Bahns T., Melzer S., God R., Krause D., 2015. Ein modellbasiertes Vorgehen zur varianten-gerechten Entwicklung modularer Produktfamilien; Tag des Systems Engineering 2015

Bursac, N., 2016. Model Based Systems Engineering zur Unterstützung der Baukastenentwicklung im Kontext der Frühen Phase der Produktgenerationsentwicklung, IPEK.

Gebhardt, N., Bahns, T., Krause, D., 2014. An example of visually supported design of modular product families. 24rd CIRP Design Conference, Milano.

Hackl, J., Krause, D., 2017. Towards an Impact Model of modular Product Structures, International Conference on Engineering Design, ICED ´17, Vol.3: Product, Services and Systems Design , Vancouver, Canada, pp. 151-160.

Hanna M., Krause D., 2017. Model-based support for product family design, in: Proceedings of the 17th International Dependency and Structure Modeling Conference, DSM 2017, Espoo, Finnland.

Herzig, S., Qamar, A., Reichwein, A., Paredis, C., 2011. A Conceptual Framework for Consistency Management in Model-Based Systems Engineering, in: Proceedings of the ASME 2011 International Design Engineering Technical Conferences &Computers and Information in Engineering Conference IDETC/CIE 2011 August 29-31, Washington, DC, USA.

Holt, J., Perry, S., Brownsword, M., 2012. Model-based requirements engineering, The Institution of Engineering and Technology, London.

Krause, D., Beckmann, G., Eilmus, S., Gebhardt, N., Jonas, H., Rettberg, R., 2014. Integrated Development of Modular Product Families: A Methods Toolkit, in Simpson, T.W., Jiao, J.R., Siddique, Z. and Hölttä-Otto, K., Advances in Product Family and Product Platform Design, Springer, New York, pp. 245-270.

Pimmler, T. U., Eppinger, S.D., 1994. Integration Analysis of Product Decompositions, in: Proceedings of the 6th Design Theory and Methodology Conference, New York, pp. 343-351

Scherer, H., 2016. Model Based Methods for the Modeling of the System of Objectives and the Correlation between Form and Function to support the Series Development of Modular Systems Using the Example of Hybrid Powertrains, IPEK.

Stachowiak, H., 1973: Allgemeine Modelltheorie, Springer, New York.

Steward, D., 1981. The Design Structure System: A Method for Managing the Design of Complex Systems, in IEE Transactions on Engineering Management, Vol. 28, No. 3, New York.

Stone, R. B., 1997. Towards a Theory of Modular Design, University of Texas, Austin.

Weilkins, T., 2008. Systems Engineering with SysML/UML: Modeling, Analysis, Design, Morgan Kaufmann.

Contact: Michael Hanna, Hamburg University of Technology, Institute of Product Development and Mechanical Engineering Design, Denickestraße 17, 21073 Hamburg, Germany, Phone +49 40.42878.3161, Fax +49 04.42878.2296, michael.hanna@tuhh.de, www.tuhh.de/pkt

Author Index