20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

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).

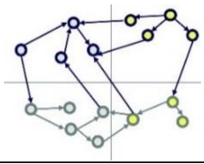
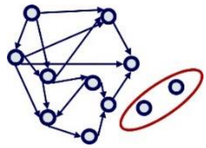
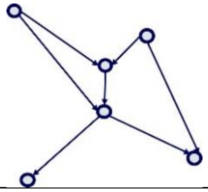
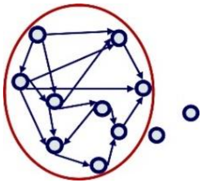
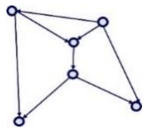
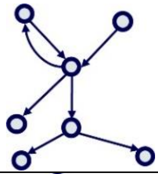
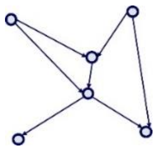
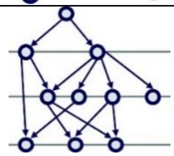
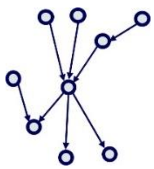
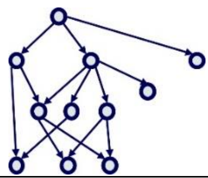
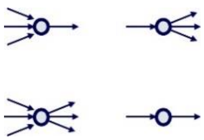
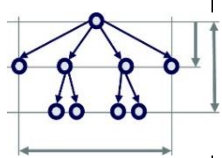
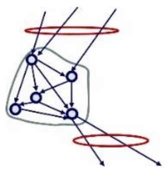
4. Deriving Structural Metrics from Collaboration Graphs

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.

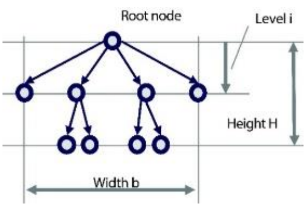


Figure 2. Snowball factor, spreading of information or errors within a network (Kreimeyer, 2009).

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

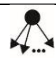
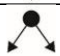
To describe the human ability to understand both particular parts of the network and how a network is structured, a metric for describing the cognitive weight is defined by Wang (2006). This metric represents the sum of the cognitive weight of each individual node that is part of the network.

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).

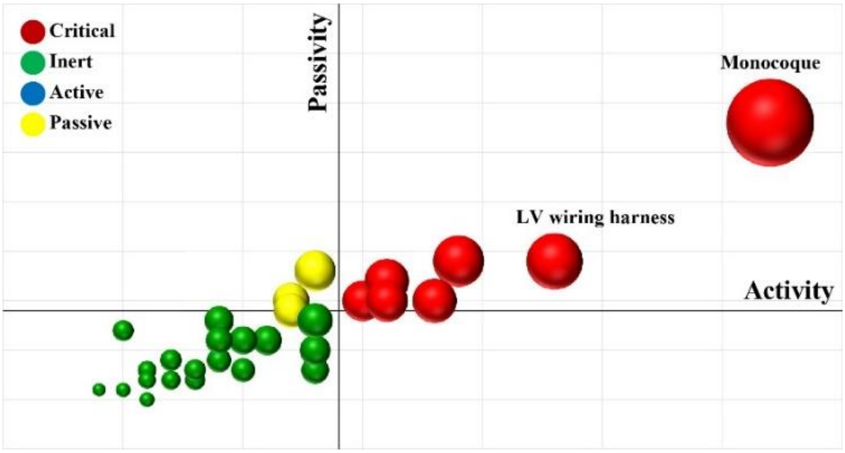


Figure 3. Influence portfolio of the components from the dataset of University Racing Eindhoven and their interdependencies.

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.

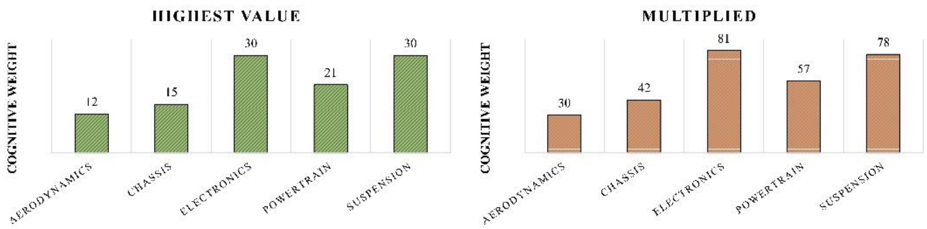


Figure 4. Cognitive weight for each of the domains, determined using the "Highest value" method on the left and using the "Multiplied" method on the right.

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