

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Produktentwicklung

A minimal set of network metrics for analysing mechatronic product concepts

Wieland Biedermann

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

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. Gunther Reinhart
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. Udo Lindemann
	2.	Prof. Peter John Clarkson, Ph.D., University of Cambridge/UK

Die Dissertation wurde am 28.05.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 10.12.2014 angenommen.

The following publications are part of the work presented in this thesis:

Biedermann, W.; Maurer, M.; Lindemann, U.: The Multiple-Domain-Approach and Cost Attributes. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): Proceedings of the 9th International DSM Conference, Munich, Germany, 16.-18.10.2007, pp. 287-295. ISBN 3-8322-6641-0.

Biedermann, W.; Lindemann, U.: Designing Consistent Structural Analysis Scenarios. In: Culley, S. J.; Hicks, B. J.; McAloone, T. C.; Howard, T. J.; Lindemann, U. (Eds.): Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 4: Product and Systems Design, Lyngby/Copenhagen, Denmark, 15.-19.08.2011, pp. 133-144. ISBN: 978-1-904670-24-7.

Biedermann, W.; Lindemann, U.: On the Applicability of Structural Criteria in Complexity Management. In: Culley, S. J.; Hicks, B. J.; McAloone, T. C.; Howard, T. J.; Lindemann, U. (Eds.): Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 4: Product and Systems Design, Lyngby/Copenhagen, Denmark, 15.-19.08.2011, pp. 21-32. ISBN: 978-1-904670-24-7.

Biedermann, W.; Lindemann, U.: Prediction of Communication Structures Based on Product Structures. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USE, 14-15 September 2011, pp. 291-300. ISBN 978-3446-43037-2.

Biedermann, W.; Lindemann, U.: Correlation of Structural Characteristics of Product Design Structure Matrices. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojcetic, N. (eds.): Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia, Vol. 3, pp. 1657-1666. ISBN 978-953-7738-20-4.

Biedermann, W.; Strelkow, B.; Karl, F., Lindemann, U.; Zaeh, M. F.: Reducing Data Acquisition Effort by Hierarchical System Modeling. In: Wynn, D. C.; Kreimeyer, M.; Eben, K.; Maurer, M.; Lindemann, U.; Clarkson, J. (eds.): Managing Complexity by Modeling Dependencies – Proceedings of the 12th International DSM Conference Cambridge, UK, 22 and 23 July 2010, pp. 309-318. ISBN: 978-3-446-42473-9.

Biedermann, W.; Diepold, K.; Lindemann, U.: Entscheidungsstrategien zur Festlegung und Priorisierung von Änderungen. In: Maurer, M.; Schulze, S.-O. (eds.): Tag des Systems Engineering – Komplexe Herausforderungen meistern, Hamburg, 9.-11. November 2011, pp. 141-150. ISBN: 978-3-446-43041-9.

Biedermann, W.; Diepold, K.J.; Lindemann, U.; Lohmann, B.: Delimiting Structural and Dynamical System Analysis in Engineering Management. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojcetic, N. (eds.): Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia, Vol. 3, pp. 1649-1656. ISBN 978-953-7738-20-4.

Diepold, K.J.; Biedermann, W.; Eben, K.G.M.; Kortler, S.; Lohmann, B.; Lindemann, U.: Combining Structural Complexity Management and Hybrid Dynamical System Modelling. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojcetic, N. (eds.): Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia, pp. 1045-1054. ISBN 978-953-7738-03-7.

Zaeh, M.F.; Reinhart, G.; Lindemann, U.; Karl, F., Biedermann, W.: Evaluating the Innovation Ability of Manufacturing Resources. In: World Academy of Science, Engineering and Technology, Vol. 72, 2010, pp. 142-148. ISSN 2010-376X.

Zaeh, M.F.; Reinhart, G.; Lindemann, U.; Karl, F., Biedermann, W.: DSM-Based Evaluation of Assembly Manufacturing Resources. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.):

Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USA, 14-15 September 2011, pp. 435-448. ISBN 978-3-446-43037-2.

CONTENTS

1. Introduction	13
1.1 Industrial application of structural analysis	14
1.2 Increasing the applicability of structural analysis in industry	15
1.3 Preservation of analytical power with minimal sets	16
1.4 Consulting and research on structural analysis	17
1.5 Meta-analysis with clustered correlation matrices	18
1.6 Thematic classification of the thesis	20
1.7 Structure of the thesis	20
2. Graph-based product component models	23
2.1 Component models of hardware	24
2.1.1 Breakdown structure	25
2.1.2 Interface structure	26
2.2 Component models of software	29
2.2.1 Nesting structure	30
2.2.2 Dependency structure	31
2.3 Context models of components	32
2.3.1 Functions and requirements	34
2.3.2 Technical parameters	35
2.3.3 Component variants and product variants	36
2.3.4 Organisations	37
2.4 Conclusion	38
3. Structural analysis of component networks	39
3.1 Fundamentals of structural analysis	40
3.1.1 Graph theory, network theory and matrices	40
3.1.2 Classification of structural analyses	42
3.2 Assessment of life-cycle properties	43
3.2.1 Robustness	46

3.2.2	Changeability	46
3.2.3	Flexibility	47
3.2.4	Adaptability	48
3.3	Assessment of product-inherent properties	48
3.3.1	Quality	49
3.3.2	Modularity	50
3.3.3	Complexity	52
3.4	Product decomposition and modularisation	54
3.4.1	Target structures for clustering	55
3.4.2	Objective functions for clustering	56
3.4.3	Clustering algorithms	56
3.4.4	Network models for component clustering	57
3.5	Description of product and component changes	58
3.5.1	Structural models describing change propagation	58
3.5.2	Empirical research on change propagation	59
3.5.3	Simulation approaches for change prediction	59
3.5.4	Design strategies for limiting change propagation	60
3.6	Conclusion	60
4.	Improving structural analyses	63
4.1	Validity criteria of structural analyses	63
4.1.1	Computability	64
4.1.2	Disparity	65
4.1.3	Significance	66
4.2	Methods for showing the significance of structure analyses	66
4.2.1	Analogy	67
4.2.2	Comparison	67
4.2.3	Simulation	68
4.2.4	Empiricism	69
4.3	Structural analysis scenarios	69
4.4	Minimal analysis sets	71
4.5	Summary	72

5. Determining the interdependencies of structural criteria	75
5.1 Taxonomy of relations among structural analyses	75
5.2 Collection of structural analyses	77
5.3 Procedure to determine metric of the minimal set sets	78
5.3.1 Research for model collections	79
5.3.2 Computations of the structural metrics	80
5.3.3 Correlation analysis of structural metrics	80
5.3.4 Determination of metrics of the minimal set sets	80
5.3.5 Validation of metric of the minimal set sets	81
5.4 Summary	81
6. Metric of the minimal set for concept analysis	83
6.1 Model collections	83
6.1.1 Literature review	83
6.1.2 Design Repository	84
6.2 Correlation matrices of the metrics	85
6.2.1 Correlations for global metrics	85
6.2.2 Correlations for local metrics	88
6.3 Metric of the minimal set sets for local and global analysis	90
6.4 Conclusion	91
7. Simulation analysis of product change dynamics	93
7.1 Simulation of product changes	93
7.2 Simulation runs and assessment	95
7.3 Structural properties of changing components	96
7.4 Power and limits of structural analysis for change prediction	100
7.5 Validation of the minimal set of local metrics	100
7.6 Conclusion	102
8. Empirical analysis of concept reconfigurations	103
8.1 Modelling of concept reconfigurations	104
8.2 Observation of production cell reconfigurations	105
8.3 Structural properties of reconfigured components	107

8.4	Prediction of reconfigurations with structural metrics	109
8.5	Validation of the minimal set	110
8.6	Conclusion	112
9.	Conclusion and Outlook	115
9.1	Metric of the minimal set sets for product concepts	115
9.2	Empirical validation of structural concept analysis	117
9.3	Structural analysis of non-product systems	118
9.4	Comparison of structural analyses to other approaches	118
10.	References	119
11.	Appendix	141
11.1	Subsets of graphs	141
11.2	Global structural metrics	142
11.3	Local structural metrics	143
11.4	Relations by definition among structural analyses	143
11.5	Models from the literature review	147
11.6	Models from the design repository	149
11.7	Modelling and simulation details for the change case study	151
11.8	Full results of the reconfiguration case study	154

FIGURES

Figure 1-1 Process of industrial application of and research on structural analysis	13
Figure 1-2 Objectives and solution requirements of this thesis	17
Figure 1-3 Research approach, applied methods and main outcomes	19
Figure 1-4 Chapters and main contents of this thesis	22
Figure 2-1 Product components in structural models – relations within the models and context of the models	23
Figure 2-2 Network models of hardware components – breakdown and interface structure	24
Figure 2-3 Network models of software components – nesting and dependency structure	29
Figure 2-4 Context of network models of product components	33
Figure 3-1 Purposes and outcomes of structural analysis of product networks	40
Figure 3-2 Classification of structural analyses	42
Figure 3-3 Enabler network of life-cycle properties based on [DE WECK ET AL. 2012]	44
Figure 3-4 Classification of product-inherent properties	48
Figure 4-1 Criteria for the applicability of structural analysis and methods to test the criteria	64
Figure 4-2 Methods for showing the significance of structural analysis	66
Figure 4-3 Contents of a Structural Analysis Scenario including a simplified example taken from [SOSA ET AL. 2007A]	70
Figure 4-4 Fundamental analysis sets	72
Figure 5-1 Dependencies of the significances due to relations among the structural analyses	76
Figure 5-2 Main activities and their outcome for determining and evaluating the metric of the minimal set	78
Figure 6-1 Correlation matrix of the global metrics for the literature model collection	86
Figure 6-2 Correlation matrix of the global metrics for the repository model collection	87
Figure 6-3 Correlation matrix of the local metrics for the literature model collection	89
Figure 6-4 Correlation matrix of the local metrics for the repository model collection	90
Figure 7-1 Modelling process for multi-dynamic mapping	94
Figure 7-2 Decision-based change simulation	95
Figure 7-3 Example scatter plots for the relations between simulations results and structural metrics	97

Figure 7-4 Scatter plots of the structural properties and change response variations	99
Figure 7-5 Comparison of the metrics in the degree cluster concerning change prediction	101
Figure 8-1 Design and reconfiguration in the life-cycle of a production resource	104
Figure 8-2 Types of production cell reconfigurations	105
Figure 8-3 Comparison of changed, removed and unchanged components for the product change at module level	108
Figure 8-4 Comparison of extended and unchanged components for the product change at module level	109
Figure 8-5 Comparison of the metrics in the degree cluster for changed, unchanged and removed components for the product change at module level	111
Figure 8-6 Comparison of the metrics in the degree cluster for extended and unchanged components for the product change at module level	112
Figure 11-1 Inheritance relations among structural analyses	144
Figure 11-2 Composition relations among structural analyses	145
Figure 11-3 Derivation relations among structural analyses	146

TABLES

Table 2-1 Levels of abstraction of hardware components	26
Table 2-2 Relations among hardware components	28
Table 2-3 Types of software components	31
Table 2-4 Relations among software components	32
Table 3-1 Structural analyses and visualisations	43
Table 3-2 Life-cycle properties according to [DE WECK ET AL. 2012]	45
Table 3-3 Product-inherent properties	49
Table 5-1 Structural metric collections	77
Table 8-1 Number of production cell reconfigurations	106
Table 11-1 Definition of the subsets of graphs	141
Table 11-2 Definitions of the global metrics	142
Table 11-3 Definitions of the local metrics	143
Table 11-4 Implications of relations by definition among structural analyses	143
Table 11-5 Models from the literature review (part 1 of 2)	147
Table 11-6 Models from the literature review (part 2 of 2)	148
Table 11-7 Models from the design repository (part 1 of 3)	149
Table 11-8 Models from the design repository (part 2 of 3)	150
Table 11-9 Models from the design repository (part 3 of 3)	151
Table 11-10 Consistency relations leading to changes	151
Table 11-11 Priority rules for change requests	152
Table 11-12 Priority rules for change options	153
Table 11-13 Average values of the metrics for each reconfiguration (changed and removed)	154
Table 11-14 Average values of the metrics for each reconfiguration (extended)	155
Table 11-15 Level of significance of the observations for each reconfiguration (changed and removed; part 1 of 2)	156
Table 11-16 Level of significance of the observations for each reconfiguration (changed and removed; part 2 of 2)	157
Table 11-17 Level of significance of the observations for each reconfiguration (extended)	158

1. Introduction

Today, most engineering projects involve several engineering disciplines such as mechanical, electrical and software engineering and result in a product family rather than single products. The main reasons are an increased demand for a large variety of functionalities and a more competitive market environment. Both reasons are driven by social, societal and technological developments like globalisation and the internet. However, these changes have made these engineering projects difficult to manage. Commonly engineering systems are said to become more complex. [WARFIELD 1999]

Managing these changes is one of the key challenges of engineering management today. Several tools have been proposed to support engineers and manager alike to better deal with engineering systems. One such tool is structural analysis, which focuses on modelling and understanding the structure of a system and its impact on the performance, design and implementation of the systems. [BROWNING 2001]

Figure 1-1 shows the process of applying structural analysis (loosely based on [LINDEMANN ET AL. 2009]). At the beginning is an engineering task such as defining modules of the product, which is to be solved by structural analysis e. g. the definition of modules. First data about the system structure such as the product components and their interfaces is acquired. Then the data is transformed into a model of the structure e.g. a design structure matrix (DSM). Last the model is analysed e.g. by clustering the matrix to find a solution for the initial task e. g. a proposal for modules.

Figure 1-1 also shows the main task of research on structural analysis: proving its feasibility. Research has to show that the result of the analysis process really solves the initial task and that structural analysis is the most efficient or the most effective way to accomplish the task.

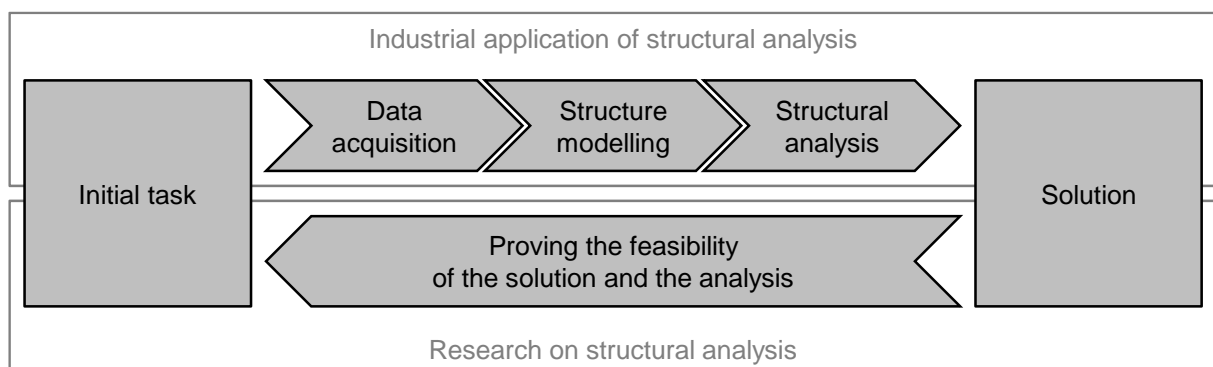


Figure 1-1 Process of industrial application of and research on structural analysis

Structural analysis is closely associated with models of the system structure. Mostly, the modelling technique is better known than the analysis technique. The best known modelling technique is the design structure matrix (DSM), which is a square matrix. The rows and columns depict the elements of the system; the non-zero entries of the matrix depict the relations among

the elements [STEWART 1981]. Other models of the structure include network or graph representations of the structure. As all of these models show the same information [HARARY ET AL. 1965] they are subsumed in the term “structural model” in this thesis.

1.1 Industrial application of structural analysis

Current research on structural analysis claims that considerations of the structure provide concise, flexible and intuitive tools for visualizing and analysing complex technical systems [EPPINGER & BROWNING 2013]. Nevertheless there is hardly any application of structural analysis in industry, which does not involve academia or specialised consultants.

Several contributions and key note speeches of practitioners acknowledge that structural modelling and analysis is a valuable tool in concept design of products and in setting up engineering projects. They point out that structural models concisely provide a lot of information on various aspects of a system. By allocating much information into one model many tasks can be solved as a wide range of analyses becomes available. They also compliment on the flexibility of structural modelling as it allows for describing many facets of a product (e.g. technical interfaces and functional allocation) at several levels of abstraction (e.g. modules and components of a product). Finally, practitioners point out the intuitive understanding of the models and the analysis results. Most engineering and manager easily grasp the content of the models and the implications of the results at first sight.

Nevertheless, most practitioners consider structural modelling and analysis not ready for widespread industrial application. Even though the models are intuitively understandable once created, the process of creating and analysing them is far too complicated and time-consuming. In particular, they criticise the lack of guidance during modelling and analysis. [MAGEE 2001], [WALDMAN 2006], [FLANAGAN 2007], [HERFELD 2008], [ROOSMALEN 2008], [CALLAHAN 2009], [EPPINGER 2009] and [WHITNEY 2011]

This seeming contradiction results from the conciseness and flexibility of structural modelling and analysis. As there is a lot of data to acquire and to process from a wide range of sources the modelling becomes time-consuming. As there are many ways to present the structure and even more ways to analyse it, the analysis becomes complicated. Both effects interact as modelling and analysis are usually iterative. That means that analysis results may trigger an update of the model, which results in a reanalysis. Only up-to-date models are prone to be accepted by practitioners [CALLAHAN 2009].

Practitioners repeatedly called for creating guidelines for structural analysis and thereby simplifying its application [HERFELD 2008], [CALLAHAN 2009]. However, most research so far focused on understanding the impact of the structure or on creating new modelling or analysis methods. Due to the high effort of creating structural models most research presents only single case studies and little consolidating research has been done. Thus, the current state of the art does not provide the data for creating general guidelines as most generalisations are based on theoretical arguments rather than empirical evidence.

As there are so many structural models and analyses it takes quite a lot of time to become familiar with them and to be able to choose the appropriate models and analyses. Some authors claim that it takes months to get acquainted with structural analysis and years to become an

expert [EPPINGER & BROWNING 2012]. Thus, there is hardly any application of structural analysis in industry, which does not involve academia or specialised consultants. Only academics and consultants have the time to train themselves in structural analysis.

From an industrial perspective structural analysis “is too complicated and too time-consuming” [HERFELD 2008] although the achieved insights and results are acknowledged. From a scientific point of view research on structural analysis has been mostly exploratory so far and has produced mostly qualitative results. Thus, the current state of the art in structural analysis fulfils neither the demand of industry for simple straightforward analysis nor the requirements of science for rigor. See also Figure 1-2 for a concise presentation of the problem description, the objectives and the solution requirements of this thesis.

1.2 Increasing the applicability of structural analysis in industry

The main objective of this thesis is to consolidate the current state of the art of structural analysis and to provide more guidance on structural analysis. This thesis focuses on structural metrics applied to structural models of product components.

The focus on metrics results from the wish for guidance in industry. Metrics are usually deterministic whereas visualisations are not. Many algorithms for visualizing networks and graphs contain a probabilistic procedure [BATTISTA ET AL. 1998]. In particular, strength-based graphs are usually not deterministic and depend on the initial layout of the network. This applies to many modularisation and clustering techniques as well [HARTIGAN 1975]. Some textbooks therefore recommend to create several visualisations and to improve them manually later. However, this does not comply with the demand for simple analysis as expressed by the practitioners. Therefore, this thesis uses metrics as the main analysis tool since their computation and presentation is usually deterministic.

To consolidate the current research on structural metrics this thesis aims at finding a minimal set of structural metrics, which covers all analyses available through structural analysis. This main aim breaks down to five sub-objectives:

- Identify main types of network models of components, the applied structural metrics and the main purposes of models and analyses.
- Define criteria for the applicability of structural metrics and defining methods for testing the criteria.
- Identify candidates for metrics of the minimal set by rigorously testing the applicability criteria for a wide range of component models.
- Verify the potential metrics of the minimal set in selected case studies by comparing their results to all available metrics.
- Define structural analysis scenarios, which comprise all information needed for performing an analysis for a given purpose.

By achieving the first objective the scope for the remaining thesis can be defined. All identified metrics are candidates for metrics of the minimal set. The identified models and purposes define the input data for the metric-based analysis and the analyses, which the metrics have to provide. The second objective is needed to define the theoretical basis for reducing the number metrics as the applicability criteria allow for excluding metrics from the minimal set. The third objective

is the main aim of this thesis as it reduces the number of metrics. However, it may result in multiple sets of candidates. The fourth objective is necessary to evaluate the main results of this thesis and to prove the relevance and significance of them. The fifth objective refers to the demand for guidance when applying structural metrics. The scenarios guide practitioners through the application of structural analyses. However, this thesis does not aim at providing analysis scenarios for all available metrics. See also Figure 1-2 for a concise presentation of the problem description, the objectives and the solution requirements of this thesis.

1.3 Preservation of analytical power with minimal sets

The main aim of the thesis is to find and verify a minimal set of structural metrics for component networks. The two main requirements for a minimal set are its minimal size and its maximal analytical power. Minimal size means that the set should contain a minimal number of metrics. Maximal analytical power means that the set should provide the same insights and conclusions as all available metrics. The metrics may have individual meaning as well as in combination. Therefore, all combinations of the metrics of the minimal set have to be taken into account when assessing their analytical power.

The significance of the minimal set is limited through the structural models it refers to. If a minimal set refers to a type of structural models, which is hardly used in practice, its significance is rather low.

Another requirement comes from metric theory: independence of the metrics. It refers to the values of the metrics and requires them to be mutually independent. All value combinations should be possible and at best equally probable. If this does not apply the metrics are at least slightly correlated and may be redundant.

All the requirements above must be fulfilled to find a minimal set of metrics. However, they may result in multiple candidates. In this case two additional “soft” requirements allow for selecting the final set: simplicity of definition and simplicity of computation. The simpler the definition the simpler the metric and its implications are to understand. The simpler the computation the faster the metric is available. Whereas the first requirement comes directly from the demand for guidance the second requirement rather refers to the tools for structural analysis. The computation requirement is hardly relevant today as advances in algorithmic graph theory allow for real-time computing. However, there are still single metrics, which take rather long to compute e.g. the cycles in a structure. See also Figure 1-2 for a concise presentation of the problem description, the objectives and the solution requirements of this thesis.

Main points of critique concerning structural analysis	
<ul style="list-style-type: none"> + Valuable tool in concept design + Models and result early available + Wide range of systems and perspectives + Basis for further analysis e.g. simulation 	<ul style="list-style-type: none"> – Modelling effort too high – Level of abstraction sometimes too high – Too many models and analyses – Little guidance in modelling and analysis
Objectives of this thesis	
<ul style="list-style-type: none"> • Identify main component network models, main analyses and the main analysis purposes • Define criteria for the applicability of structural analyses and methods for testing the criteria • Define structural analysis scenarios which comprise the information for performing an analysis • Find a minimal set of structural metrics • Consolidate the existing research on structural analysis 	
Requirements for the minimal set of structural metrics	
<ul style="list-style-type: none"> • Minimal number of metrics which still provides the analytical power of the available metrics. • Valid for a relevant family of structural models of product components. • Mutually independent metrics i.e. the values are not correlated. • Allow for the same conclusions in the relevant contexts as the available metrics. • Simplicity of definition and computation 	

Figure 1-2 Objectives and solution requirements of this thesis

1.4 Consulting and research on structural analysis

As stated above it takes years to become an expert on structural analysis. The author has worked for the past ten years in the area of structural analysis both in industrial and academic environments. Thereby, he became familiar with both the application and the theoretical foundations of structural analysis. However, his main field of expertise is the analysis part rather than the modelling part.

From 2003 to 2007 the author worked as a student assistant at the Institute of Product Development at the Technische Universität München. During this time he specified and tested a software tool for structural analysis with a focus on complexity management. Later he coordinated the development of the tool. The tool is today marketed as Loomeo by the Teseon GmbH. During his work the author became familiar with the theoretical and algorithmic foundations of structural analysis. As he was also involved in several research projects he also got first insights into the application of structural analysis. His main observation was the trend of reinvention in structural analysis research due to a lack of knowledge in graph theory.

From 2007 to 2009 the author worked as developer and consultant in the area of complexity management. He was part of several projects e.g. on variant management or quality processes in automotive or on airport security. Structural analysis was part of all projects. Therefore, the author became familiar with structural analysis in industrial application. His main insight was

that in most cases the simplest analysis with a clear interpretation provided most of the benefit to industry.

From 2007 to 2012 the author worked as a research assistant. His main research project dealt with dependency modelling and was part of the collaborative research centre SFB 768. During this time he developed the ideas and results presented in this thesis. The nature of SFB 768 provided him with research opportunities in several engineering disciplines. Thus, he became familiar with a wide range of applications of structural analysis. His main observation was a lack of consolidation in research on structural analysis due to a lack of data and the high modelling effort.

Through the work of the past ten years the author acquired the expertise to write this thesis. Particularly the possibility to get both insights from the academic as well as the industry side was important.

1.5 Meta-analysis with clustered correlation matrices

This research follows loosely the design research methodology described in [BLESSING & CHAKRABATI 2009] (and is most closely associated with project type 3). Figure 1-3 gives an overview over the main research methods and the main outcomes on this research.

The research clarification is based on a literature review and the experience of the author. The literature review focused on case studies and key note speeches by practitioners in industry. Most of them are closely linked to the DSM community and have a research background on structural analysis. A thorough analysis of their contributions resulted in the problem description, objectives and solution requirements above (see sections 1.1 to 1.3).

The descriptive study I was derived from an extensive review of the literature on structural analysis especially the design structure matrix. The main sources of literature were DSMweb, a keyword search and a review of major journals. DSMweb is the online platform for the DSM community and provides subscribers with 1100 publications (as of January 10, 2013) on dependency and structure modelling. To extent the sources a keyword search was performed, which focused on the years from 2008 to 2012 and used keywords like DSM, DMM, etc. The review on sixteen major journals focused on the years from 2001 to 2011. The journals were chosen based on their ISI rating and were drawn from the fields of engineering design, engineering management and systems engineering (see [BIEDERMANN & LINDEMANN 2012] for a detailed description). This resulting collection of publications was then filtered to include only publications, which describe the structural modelling and analysis of products modelled as component networks. From the analysis of the remaining publications the available and commonly used structural models, the available structural metrics and the common purposes of structural analysis were derived. However, a quantitative analysis of the literature was omitted as the qualitative results suffice for the sake of this thesis. The findings on structural product models are presented in chapter 2. The findings on structural analysis are to be found in chapter 3.

In the prescriptive study the metric of the minimal set is derived by comprehensive analysis of structural models of product components. The theoretical foundation for the derivation of the minimal set is a collection of criteria for the applicability of structural metrics. The main

criterion for this thesis is disparity whereas the most important criterion for the applicability is significance. Chapter 4 describes the criteria and testing methods in detail. Disparity is tested by determining interdependencies among the structural metrics. One of the most important types of interdependencies is correlation, which is determined by a correlation analysis of structural metrics for a set of structural models of the same type. Chapter 5 presents the method for determining the minimal set in detail. In this thesis two sets are used. The first set results from a literature review and determines the initial candidates for the minimal set. The second set results from an online repository and validates the findings from the first set. The detailed results of the correlation analysis and the final minimal set are shown in chapter 6.

Research phase	Type	Main method	Main outcomes
Research clarification	Literature based	Literature review	<ul style="list-style-type: none"> • Advantages/disadvantages of structural analysis • Industrial/academic perspectives • Objectives of the thesis • Requirements for the solution
Descriptive study I	Literature based	Literature review	<ul style="list-style-type: none"> • Available types of structural product models • Commonly used types of structural product models • Available structural analyses • Main purposes of structural analysis in engineering
Prescriptive study	Comprehensive	Meta analysis	<ul style="list-style-type: none"> • Taxonomy of relations among structural analyses • Definition of structural analysis scenarios • Two sets of component models • Two minimal sets of structural metrics
Descriptive study II	Initial	Case studies	<ul style="list-style-type: none"> • Two new analysis scenarios • Validation of the two minimal sets

Figure 1-3 Research approach, applied methods and main outcomes

The descriptive study II comprises two case studies, which validate the minimal set via simulation of changes and empiricism on production reconfigurations. The idea behind the case studies is to use the minimal set to analyse product models. Then the models are analysed with all available metrics. Finally, the results are compared. If the minimal set allows for the same conclusions as all metrics it is verified. Otherwise it has to be rejected. Both case studies were conducted in the SFB 768. The first case study deals with change simulation and tests if simulation results can be predicted by structural analysis. It was done in cooperation with the Institute of Automatic Control of Technische Universität München. The second case study deals with the reconfiguration of production resources. It compares the values of structural metrics for changed, removed, extended and unchanged components. The study was done in cooperation with the Institute for Machine Tools and Industrial Management of Technische Universität München. The detailed research methodology is presented in the respective

chapters. Chapter 7 gives the results on changes. Chapter 8 presents the results on production reconfiguration.

1.6 Thematic classification of the thesis

This thesis contributes to several research areas and engineering disciplines. Therefore, several classifications are sensible.

First, the thesis contributes to the areas of product architecture design, product concept analysis and complexity management as structural analysis is part of each research area. The thesis uses input from graph and metrics theory. It contributes to three engineering disciplines: systems engineering, mechanical engineering, production engineering – other areas such as software engineering may benefit as well. The contribution to product architecture design directly impacts systems engineering by consolidating the available structural metrics for product concept analysis. The same holds true for product concept analysis and mechanical engineering. The main contribution to production engineering is the case study on production reconfiguration.

Second, the thesis aims at improving the activities during the concept phase of the product life cycle. Researchers claim that structural analysis is one of the first methods available for concept assessment as it requires only data, which is available early in the design phase. Though later phase may benefit as well from structural considerations more sophisticated quantitative methods are then available, which provide more detailed results.

Third, the thesis focuses on the structural analysis of products and, in particular component networks. As stated above structural models are very flexible and allow for modelling a wide range of systems e.g. process and organisation architectures. Though the research methods can be transferred the results in particular the minimal set only applies to component networks.

Fourth, the thesis primarily addresses the analysis of structural model not their creation. In past research model creation and analysis were usually considered in combination. However, model creation and model analysis should be separated. According to general model theory the mode of creation should not influence the characteristics of the model and therefore not influence the analysis results. This thesis assumes this to be correct even though some authors suggest otherwise.

Fifth, this thesis mainly contributes to research on structural analysis by defining applicability criteria and providing a method to test them. The other results such as the minimal set only apply to a limited set of models (namely component networks). Though this eases the application of structural analysis in industrial practice the challenges concerning model creation have to be resolved in future research.

1.7 Structure of the thesis

This thesis comprises three major parts. The first part (chapters 2 and 3) present the state of the art on structural analysis of product component networks. The second part (chapters 4 to 6) describes the theory, derivation and first validation of the metric of the minimal set. The third part (chapters 7 and 8) presents two case studies for the validation of the metric set in

applications. The chapters 1 (introduction) and 9 (conclusion) form the framework for the remaining thesis. Figure 1-4 shows the complete structure of the thesis including the main contents of each chapter and the question to be answered in the chapters.

The first part of the state of the art (chapter 2) describes the models used for depicting product component networks. Though it focuses on hardware products (section 2.1) the chapter also describes the fundamentals of structural software models (section 2.2). As the components never stand alone in any product or engineering project the chapter also discusses the contexts of the components namely requirements, functions, technical parameters, variants and organisations (2.3). The second part of the state of the art (chapter 3) addresses the analysis of structural models. After the discussing the theoretical background of structural analysis (section 3.1) the chapter discussed the main purposes of analysing components networks (sections 3.2 to 3.5). In particular the sections 3.2 and 3.3 deal with metric based analysis, whereas the remaining sections put the analysis into a wider context.

The core chapters of this thesis describe the creation of the metric of the minimal set. Chapter 4 presents structural analysis scenarios and metric of the minimal set sets as the main approaches to overcome the lack in simplicity and rigor of structural analysis. First the criteria for the applicability (section 4.1) and the corresponding testing methods (section 4.2) are described. After that the main contributions of this thesis are introduced: analysis scenarios (section 4.3) and metric of the minimal set sets (section 4.4). Chapter 5 shows the research methodology to identify metric of the minimal set sets in detail. After presenting the relations among metrics (section 5.1), which allow for identifying minimal sets and the metrics (section 5.2) researched in this thesis the chapter presents the procedure for determining the minimal set (section 5.3). Chapter 6 presents the core results of this research – in particular the metrics of the minimal sets for analysing whole concepts and for analysing elements within a concept. First the chapter describes the two model collections used for deriving the metrics (section 6.1). Then it presents the major intermediate result: the correlation matrices (section 6.2). Finally, it presents the proposed metric of the minimal set for analysing product component networks (section 6.3). The initial verification of both the correlation matrices and the minimal sets is presented in the two sections as well.

The two case study chapters present the initial validation of the metrics of the minimal set. The first case study (chapter 7) deals with the simulation of production changes and tests if the simulation results can be predicted by structural analysis. The second case study (chapter 8) is about production reconfiguration and shows that structural analysis allows for identifying components, which are likely to be changed.

The final chapter concludes the thesis and puts its result in a broader context. In particular it discusses if the results can be generalised to other models and domains.

1	<u>Introduction:</u> structural analysis	<ul style="list-style-type: none"> • What is structural analysis? • What needs to be improved?
2	<u>State of the art:</u> Structural product models	<ul style="list-style-type: none"> • What models are available? • Which models are commonly used?
3	<u>State of the art:</u> Structural analysis in concept design	<ul style="list-style-type: none"> • How are structural models analyzed? • What is derived from the results?
4	<u>Approach:</u> Structural analysis scenarios and minimal criteria sets	<ul style="list-style-type: none"> • How to simplify structural analysis? • How to increase rigor in structural analysis?
5	<u>Methodology:</u> meta analysis and case studies	<ul style="list-style-type: none"> • How are minimal metrics derived? • How are minimal metrics validated?
6	<u>Results:</u> minimal metric sets for component networks	<ul style="list-style-type: none"> • Which metrics are minimal for component networks?
7	<u>Case study:</u> Simulation of change decisions	<ul style="list-style-type: none"> • Do structural analyses predict simulations? • Are the metrics of the minimal sufficient?
8	<u>Case study:</u> Reconfiguration of production resources	<ul style="list-style-type: none"> • Do structural analyses assess reconfig.? • Are the metrics of the minimal sufficient?
9	<u>Conclusion:</u> Implications and limitations of minimal criteria sets	<ul style="list-style-type: none"> • What is improved by this thesis? • What needs to be improved further?

Figure 1-4 Chapters and main contents of this thesis

2. Graph-based product component models

This chapter presents the main models for describing product structures. As this thesis focuses on components networks this chapter is centred on product components. Coming from a mechanical engineering background, models of hardware components are presented first (section 2.1). Due to the rise of software functionalities in today's products, software components shall not be neglected and are presented next (section 2.2). In particular, the sections show, how the different natures of hardware and software are reflected in the models. Both sections describe the modelling of the vertical and lateral relations among the components. For this thesis the hardware components and their lateral relations are the most important models as they are used throughout the remaining chapters. The next section (2.3) shows the context models of the product components i.e. requirements, functions, parameters, variants and organisations. The mapping between components and their context allows for deriving relations among components and is central for documenting the design rationale and the set-up of engineering projects. This chapter focuses on describing the content of the structural models rather than their creation. Figure 2-1 shows the structure of this chapter and the relation among the subsections. For the sake of simplicity visualisations are omitted.

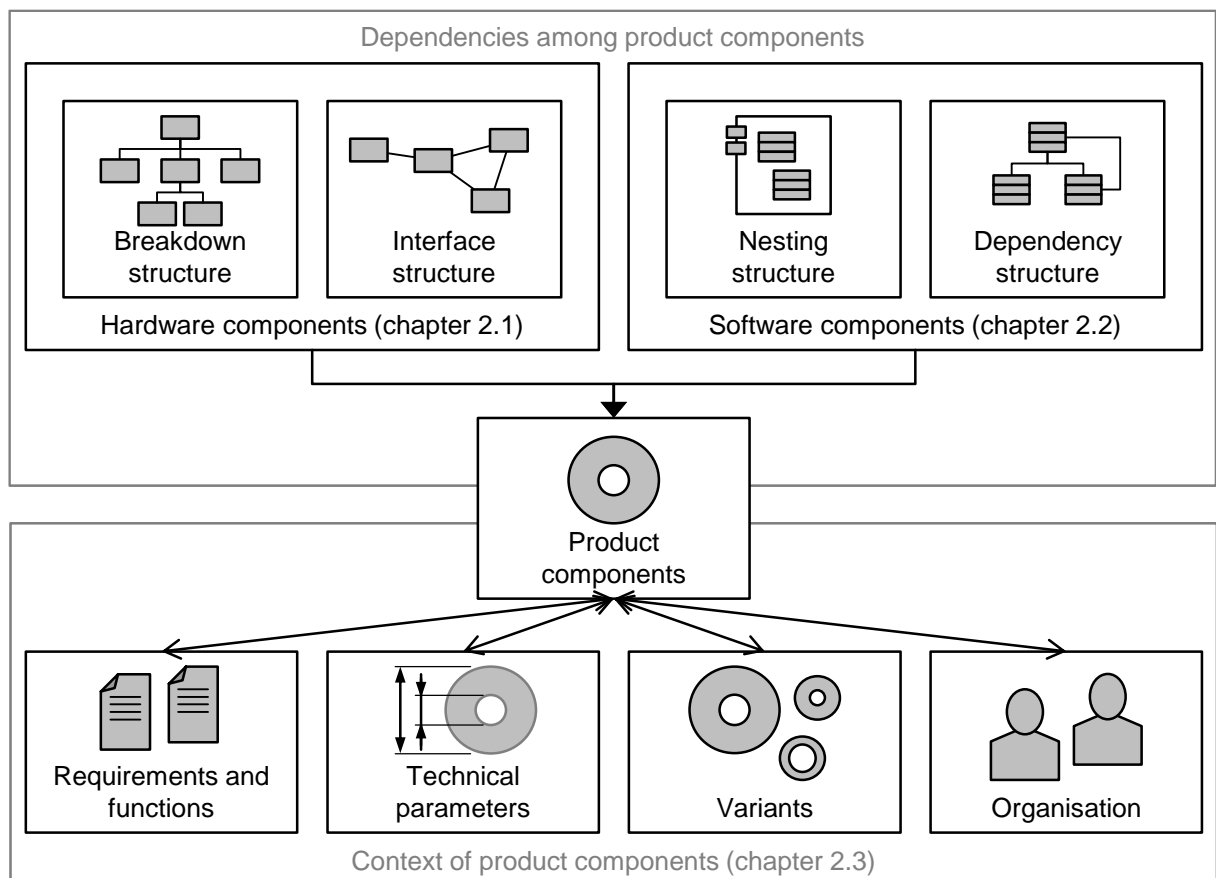


Figure 2-1 Product components in structural models – relations within the models and context of the models

2.1 Component models of hardware

Structural models of hardware components describe the physical structure of the product [ULRICH 1995]. Usually, two main structures are differentiated: the breakdown structure and the interface structure [MALMQVIST 2002]. The breakdown structure describes how the product subdivides into subsystems and modules, which in turn may subdivide into components and parts. The interface structure describes the interaction of the components, which result in interfaces. Usually, structural models (in particular the DSM) do not differentiate between components at multiple levels of abstraction (see e.g. [PIMMLER & EPPINGER 1994]). However, most authors recommend modelling the elements at the same level of abstraction. Therefore, the next subsection discusses breakdown structures, hierarchies and abstraction. The relations among hardware component usually model physical phenomena and derive their model properties (e.g. directedness) from them. Many authors simply model a general “can-change-relation”. Yet, some authors provide strong arguments for breaking this type of relation down to more detailed level (e.g. contact) to improve the model accuracy [BLOEBAUM & ENGLISH 1999]. Therefore, this chapter’s second subsection (2.1.2) provides details on interface structures. Figure 2-2 gives an overview of the levels of abstractions and the types of relations, which are described in the next subsections.

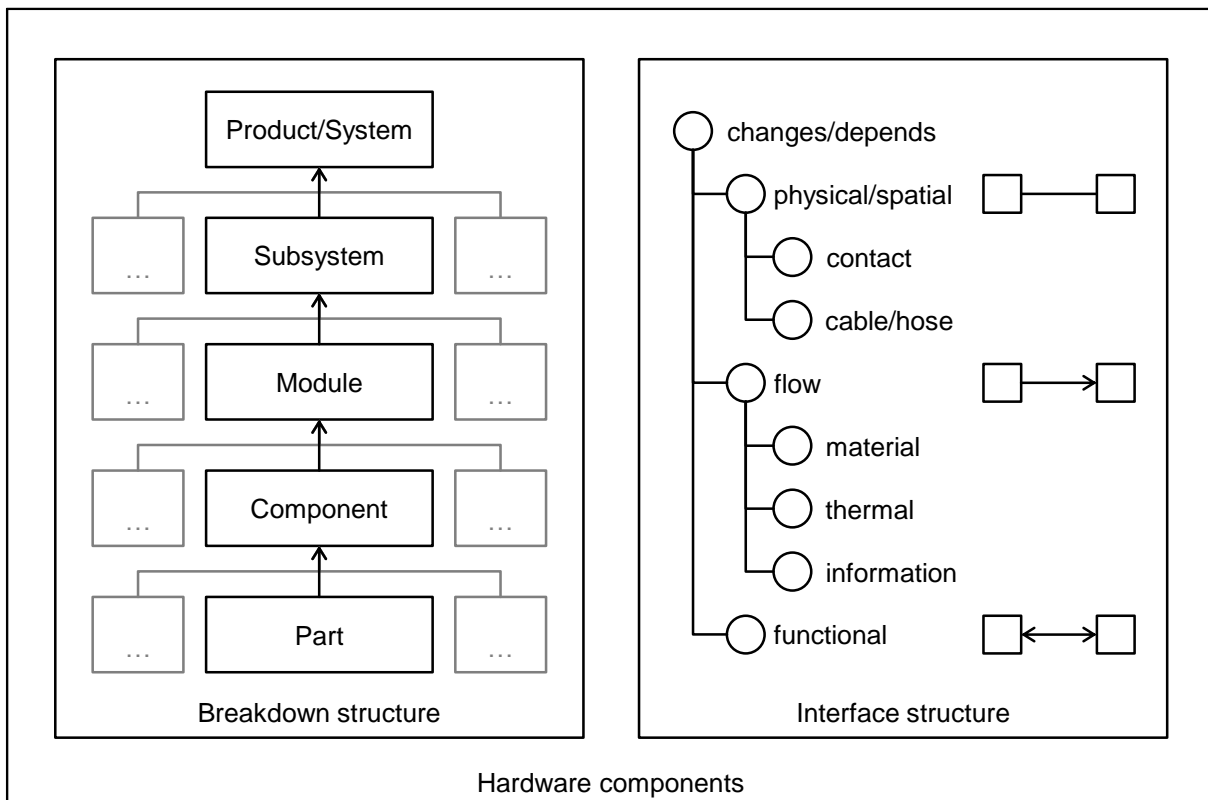


Figure 2-2 Network models of hardware components – breakdown and interface structure

Structural models of the hardware components are usually created in workshops and interviews rather than by model transformation or data mining. The main reasons are that most of models

rely on implicit knowledge and that few other types of models (e.g. CAD or simulation models) uses the same data explicitly. The creation of awareness about the data is considered as one of the most important indirect benefits of structural modelling [EPPINGER & BROWNING 2013]. However, it is also the reason for the high time demand of model creation. The improvement of the modelling process is a current field of research but not part of this thesis.

2.1.1 Breakdown structure

As stated above there is a general assumption that hardware have “natural” levels of abstraction. Several contributions define those levels and many authors suggest creating a standard for those levels to be used in academia. However, no suggested definition is generally accepted and most publications opt for component as a generic term. This thesis loosely follows the definitions by the INCOSE handbook [INCOSE 2011]. It defines five levels of abstraction (see Table 2-1). The main criterion is the artefact’s contribution to the product functionality. Systems fulfil their purpose and functionality independently whereas modules need additional facilities and components merely contribute to the system functionality.

Though such definitions are available, most researcher and practitioners use the term component to address the basic elements of their models. Thus, the term component is relative and depends on the scope and content of a model. If a model describes the public transportation system of a city any individual automobile may be considers as component or even part of the systems. The same automobile is a system to its designer and manufacturer.

If a publication explicitly states the level of abstraction, it usually deals with multiple levels, which need to be distinguished (see e.g. [ARIYO ET AL. 2007]). One classical application is modularisation. Here, the elements of the structural models are to be grouped into modules. However, mostly generic terms like “component” and “module” are used (see also section 3.4 for a detailed description of modularisation). [CHIRIAC ET AL. 2011A] show that breakdown structures differ depending on the approach for their definition.

The assumption of “natural” levels of abstraction raises the question, whether the characteristics of the component structure vary between the levels. This question has been hardly discussed in literature so far. Nevertheless, it is an important question as a variation between the levels of abstraction would strongly limit the validity of findings on structural analysis. [CHIRIAC ET AL. 2011B] provide an example for differing characteristics depending on the level of abstraction. Due to the versatile application of the term component in literature a meta-analysis of the current findings would require much effort. This thesis presents a comparison between two levels of abstraction (namely module and component level according to Table 2-1) in chapter 8. However, the results are ambivalent and do not allow for drawing a final conclusion or for stating an educated guess.

Table 2-1 Levels of abstraction of hardware components

Level	Definition	References
Product/System	Technical artefact which independently fulfills a purpose and provide an intended functionality. Example: assembly cell.	[Lapp & Golay 1997], [Bauer & Maurer 2011]
Subsystem	Product/system which contributes to the functionality of a bigger product/system. Example: welding robot.	[Lapp & Golay 1997], [Bauer & Maurer 2011]
Module	Technical artefact which fulfills a purpose and provides functionality but needs other modules/system to function. Example: robot arm.	[Ulrich 1995], [Brady 2002]
Component	Technical artefact which contributes to the functionality of a module/system but has no functionality on its own. Example: welding gun.	[Ulrich 1995], [Pimpler & Eppinger 1994]
Part	Technical artefact which is part of a component but carries no functionality: Example: screw.	[Biedermann et al. 2011]

2.1.2 Interface structure

The previous section described the breakdown structure for hardware components. This section focuses on the interface structure. Contrary to the breakdown structure the interface structures plays an important role in structural analysis as they form the structure to be analysed. The focus on the interface structure origins from the insight that they mainly determine the system behaviour and performance or as [RECHTIN 1991] put it: “the greatest leverage in systems architecting is at the interfaces.”

Most publications do not explicitly state the type of relations modelled in the component network. Usually they only refer to the DSM as a model type or state a generic type of relation like “interdependency” or “can-change-relation” (proposed e.g. by [ULRICH 1995]). Though this approach allows for very flexible modelling and captures most relations, it may lead to several problems in practice. First, it may lead to lengthy discussions about the nature of a relation as many people and disciplines are usually involved, which have their own understanding of a change (see e.g. [BLOEBAUM & ENGLISH 1999]). Second, it does not clearly differentiate between direct and indirect relations. Most authors recommend focusing on direct relations otherwise the models become less intuitive or even worthless as everything is connected to everything else (see e.g. [ALIZON ET AL. 2006]). Third, it does not allow for deciding whether a relation is directed or not. The directedness of a relation directly influences

the available analyses as a large number of analysis tools only apply to directed networks or vice versa.

To overcome these obstacles [PIMMLER & EPPINGER 1994] introduced a classification of relations using four classes namely spatial, energy, information and materials. Moreover, they introduced a second classification of relations, which bases on the contribution to the overall system functionality. They define five levels of contribution: required, desired, indifferent, undesired and detrimental. Many subsequent publications have used this set of relations to model the interfaces among the components.

Several authors modified the set of relations to allow for specific analyses or to account for specific characteristics of a system. [KELLER ET AL. 2009] extended the classification by the temporal behaviour of the interfaces. They differentiated dynamic and static interfaces for mechanical, thermal and electrical phenomena and added the relation “electrical earth”. Thereby, they captured the dynamic processes in a Diesel engine. [BIEDERMANN ET AL. 2010] subdivided the spatial relation into contact and cable relations. Both relations may lead to change propagation when modifying a component. The main difference lies in the change dynamics. Contact relations propagate a change directly whereas cable relations may have a buffer. [BIEDERMANN ET AL. 2007] used functional relations. [TILSTRA ET AL. 2009] propose a general scheme for refining structural models including the types of relations.

However, no extension was widely used. Therefore, the set by [PIMMLER & EPPINGER 1994] still is most commonly used. However, most publications only use a single relation without defining it formally. If the relation is defined the most common types are the generic change relation and the spatial relation, which is sometimes also called physical relation.

Table 2-2 shows the classification of relations used in this thesis and gives a definition for them. As Figure 2-2 shows the relations form a hierarchy. The most general type is changes/depends, which comprises all other types. The next level comprises the relations physical/spatial, flow and functional. They differ not only in their content but also in their structural characteristics: the spatial relation is undirected, the flow relation is directed and the functional relation is bidirectional. As stated above, this determines the available analyses. As the analysis for directed and undirected relations (bidirectional relations can be treated as undirected) are not compatible, the results should presented and discussed separately. The relations on the lowest level (contact, cable/hose, material, thermal, information) share the characteristics of their superior relations. The electrical relation is not included as it is hardly used in literature and can be subsumed to the flow relation.

As each additional relation creates effort during the creation of the models [BLOEBAUM & ENGLISH 1999] it would be sensible to avoid relations, which can be derived from others or, which provide no additional insights. So far there has been no research on the independence of the relations. Though [BIEDERMANN & LINDEMANN 2011C] compared three types of relations they could only show that the contact relations provided the better results than energy and information relations. They could not show that contact relations comprise the other two. Therefore, only theoretical arguments can be presented here (see also [ROCCO ET AL. 2011] for another perspective on the topic). As the relations usually model physical phenomena the relations are coupled if the phenomena are coupled. For example the thermal flow relation may require a contact relation if the heat is transferred via conduction. Alternatively it may require

a material flow if the heat is transferred via convection. The remaining potential couplings are omitted here for the sake of brevity. However, the conclusion is that there is no general rule for coupling among the relations. If each relation is only related to one phenomenon coupling rules can be defined e.g. information flows require contact or cable relations if all signals are sent electrically. This allows for cross-checking the models of the relations for consistency.

Table 2-2 Relations among hardware components

Relation	Definition	References
Changes/depends	The components are connected.	[Ulrich 1995], [de Weck 2007]
Physical/spatial	The components are physically connected or share a geometric interface.	[Pimmler & Eppinger 1994]
Contact	The components are (at least temporarily) in touch.	[Johannesson & Söderberg 2000], [Biedermann et al. 2011]
Cable/hose	The components are connected via a cable or hose.	[Lapp & Golay 1997], [Biedermann et al. 2011]
Flow	One component transfers or sends something to the other component.	[Pimmler & Eppinger 1994], [Alizon et al. 2006]
Material	One component transfers material to the other component.	[Pimmler & Eppinger 1994], [Alizon et al. 2006]
Thermal	One component transfers heat to the other component.	[Pimmler & Eppinger 1994], [Alizon et al. 2006]
Information	One component sends a signal to the other component.	[Pimmler & Eppinger 1994], [Alizon et al. 2006]
Functional	The components interact to perform a function of the system.	[Biedermann et al. 2007]

As most analyses do not account for multiple types of relations there has been some research on combining the relations and/or the analysis results to form an overall result. The simplest way is to add all relations and form a generic relation, which is then analysed. The main drawback of this solution is the loss of information during the addition process. Moreover, the contribution of each type of relation is hardly traceable. Another approach is the weighing of the relations to another. Several solutions have been proposed (e.g. [PIMMLER & EPPINGER 1994], [SHARMAN 2001], [BRADY 2002] and [HELMER ET AL. 2010]). Another approach was to add the relation is to analyse each type of relation individually and to combine the results afterwards. This approach has similar drawbacks as the relation addition. Finally, both approaches can be combined: one type of relation of relation is analysed then another type of

relation is added to the model and the analysis is updated and so on (see e.g. [GREINER ET AL. 2007]). This approach requires that one type of relation is particularly important to the analysis. In summary no approach for combining several types of relation can be recommended.

2.2 Component models of software

Structural models of software components describe the logical and behavioural structure of the product. Usually two main structures are differentiated: the artefact structure and the dependency structure. The artefact structure defines the organisation of the software in terms of functionality, deployment and development. One key characteristic of software systems is that its artefacts can be nested arbitrarily and are nested according to the requirements of the design task. Therefore, there is no natural hierarchy among software artefacts [MALMSTRÖM & MALMQVIST 1998]. The next subsection (2.2.1) deals with the nesting structure among software artefacts and defines the most important artefacts. The dependency structure describes the logical relations among the software artefacts. These relations are not bounded by the laws of physics. They are only limited by the laws of logics and mathematics. Subsection 2.2.2 describes and defines these relations in detail. Figure 2-3 gives an overview over the main software artefacts and dependencies, which are considered in structural analysis.

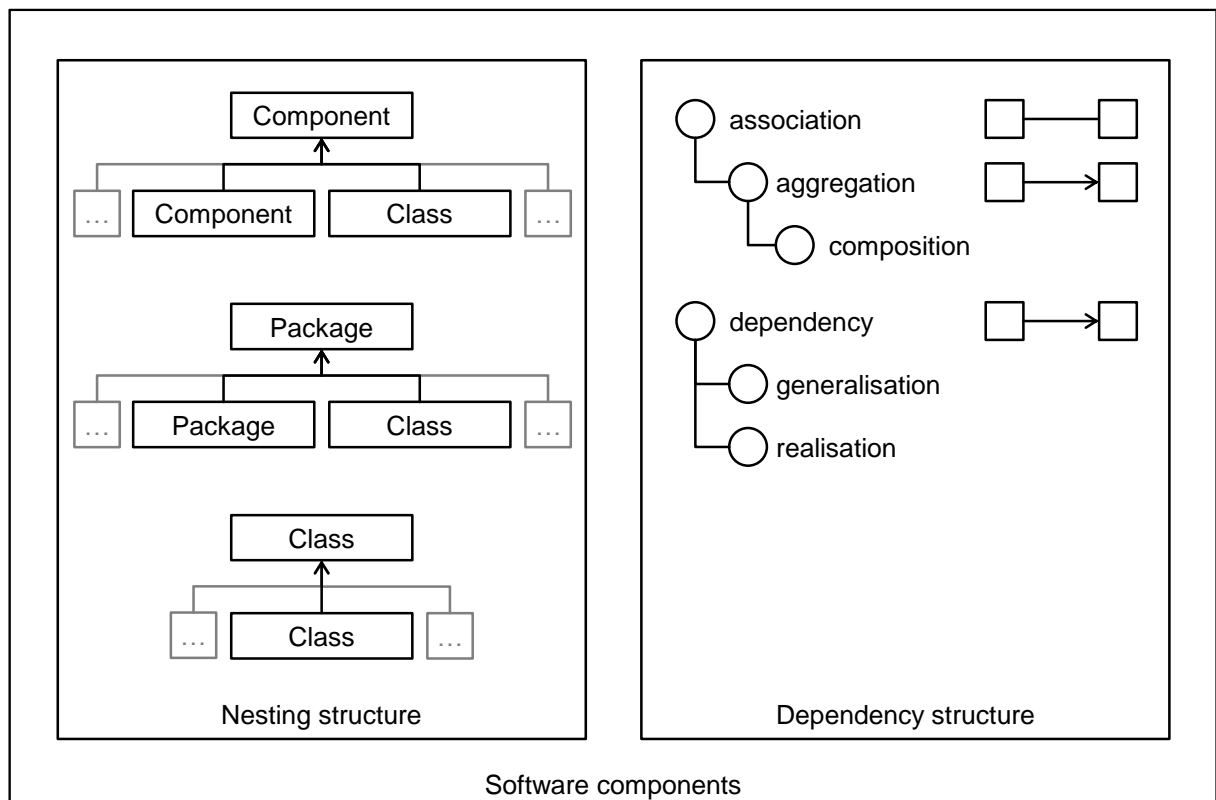


Figure 2-3 Network models of software components – nesting and dependency structure

There are three main differences between software and hardware components. First, it is generally assumed that hardware components have “natural” levels of abstraction, which are

mainly defined by their contribution to the overall system functionality. Second, the design of hardware components is strongly limited by the laws of physics. Third, the design models usually are not directly derived from the real system but are created through measuring and/or cognitive processes. All three points differ for software models and strongly influence the content of the models.

Contrary to models of the hardware, structural models of software are usually created automatically. There are two main reasons for that. First, software and its development models are very formal. Therefore, scripts and software for creating structural models can be programmed comparatively easy [DO & CARIGNAN 2005]. Second, the main outcome of software design is source code, which is computer readable by definition and allows for simple parsing to create structural models [SANGAL ET AL. 2005]. Moreover, modern programming paradigms focus on model-based development, which creates most of the software artefacts as source code rather early [DO & CARIGNAN 2005]. Therefore, the structure is also early available to automatic parsers. Together the creation of structural models of software is rather easy and fast even when modelling huge systems of several thousand artefacts [SANGAL & WALDMAN 2005].

The ease of model creation allows for extensive studies on the structure of software systems or as Browning put it: “Software systems are the fruit flies of system architecture research.” The open source movement made thousands of software systems available to research. Among the best-known examples are the operating system Linux and the web browser Mozilla Firefox. Though several publications deal with software structure the significance of their findings for hardware structures is still unknown. So far no study could conclusively show the transferability of research results from software to hardware structures. Only theoretical arguments exist.

As stated above there are three main differences between hardware and software systems when considering their structure. First, software structures have no concept of “natural” levels of abstraction. Second, relations among software artefacts mainly describe logical dependencies. Third, software structure can be derived automatically by parsing the source code. All three differences result from the nature of software as it is a virtual artefact, which is only limited by the laws of logics and mathematics. Software in all its instances (mainly development models, source code and binaries) is very formal and can be processed by computers. In contrast hardware must comply with the laws of physics.

2.2.1 Nesting structure

Software artefacts do not have “natural” levels of abstraction [MALMSTRÖM & MALMQVIST 1998]. Rather, they can be nested arbitrarily. In practice the nesting of software artefacts is a design decision and follows the requirements of the software.

Though there are many software artefacts, only three are have been commonly used in structural analysis: components, packages and classes. Classes are the basic design artefact in modern software development as the predominant design paradigm is object orientation (and its successors). They are used to encapsulate all data, it processes, and all functions, it performs, during the runtime of the software. During runtime instances of the classes (so called objects) are created and provide the software functionality. Packages are a mechanism for organizing

the software development. They contain classes and other packages and group the software artefacts according to the requirements of the development project e.g. by grouping classes with similar functionality. Components are the smallest software artefacts, which can be individually distributed. They group classes and sometimes other components, which have related functions, data objects, interfaces and data to provide functionality. Components interact via interfaces. Components and packages may use the same nesting structure but usually the two structures are not interrelated. To sum it up: packages organise the development and components organise the distribution of software. Table 2-3 gives short definitions for the main artefacts.

Table 2-3 Types of software components

Type	Definition	References
Component	A software package, a web service, or a module that encapsulates a set of related functions (or data) which may be individually distributed.	[MacCormack & Rusnak 2004], [deSouza & Redmiles 2005]
Package	A mechanism for organizing classes and software libraries.	[MacCormack & Rusnak 2004], [deSouza & Redmiles 2005]
Class	A construct that is used to create instances of itself and encapsulates all data and all functions.	[MacCormack & Rusnak 2004], [deSouza & Redmiles 2005], [Akaikine 2010]

2.2.2 Dependency structure

Contrary to the relations among hardware components, the relations among software artefacts mainly describe the logics of the interactions rather than physical phenomena. There are two main types of relations in software: association and dependency.

An association allows one software artefact to cause the other to do something. In other words one artefact calls a function of the other. There are two special types of associations (aggregation and composition), which are closely linked but do not cover all associations. In an aggregation one artefact is part of the other but exists independently. For example it can be transferred to another container. In a composition one artefact is part of the other and is not independent. It ceases to exist if the container does not exist anymore. In both cases the container can use all functions of the contained artefacts.

Dependencies are more general relations than associations as they are not limited to calling functions. If one artefact depends on the other it uses the other at some point during its life-cycle. The most important dependencies are generalisation and realisation. The generalisation relation results from the inheritance among software artefacts, which is a powerful mechanism for reusing software functionality. The parent artefact generalises one or more child artefacts. The child must provide all data and all functions of the parent and may provide additional or

more specific data and functions. The realisation relation means that one artefact implements or executes the behaviour that the other specifies. This is a strong tool for testing software.

All relations among software artefacts require an explicit statement in the source code. As these statements follow formal rules defined by the programming language, relations can be easily determined by parsing the source code.

Table 2-4 Relations among software components

Relation	Definition	References
Association	The instance of one artefact can cause another to perform an action on its behalf.	[MacCormack & Rusnak 2004], [deSouza & Redmiles 2005], [Brøndum & Zhu 2010]
Aggregation	One artefact contains other artefacts, which exist independently of the container class.	[MacCormack & Rusnak 2004], [deSouza & Redmiles 2005], [Brøndum & Zhu 2010]
Composition	One artefact contains other artefacts, which cannot exist without the container class.	[MacCormack & Rusnak 2004], [deSouza & Redmiles 2005], [Brøndum & Zhu 2010]
Dependency	One artefact depends on another because it uses it at some point of time.	[MacCormack & Rusnak 2004], [deSouza & Redmiles 2005], [Brøndum & Zhu 2010]
Generalisation	One of the two classes (the subclass) is considered to be a specialised form of the other (superclass).	[MacCormack & Rusnak 2004], [deSouza & Redmiles 2005], [Brøndum & Zhu 2010]
Realisation	One artefact implements or executes the behavior that the other artefact specifies.	[MacCormack & Rusnak 2004], [deSouza & Redmiles 2005], [Brøndum & Zhu 2010]

2.3 Context models of components

Components of technical systems always exist in a multi-faceted environment [SULLIVAN ET AL. 2001]. Through this environment additional relations among components may emerge. For example, wheels and engine of an automobile do not have as physical interface but they are linked by their contribution to driving. These relations are described in this section as they are not covered by the previous sections.

As these relations are indirect and arise from the context, they are often modelled by describing the mapping between components and the entities in the context. These mappings are often captured in domain mapping matrices (DMM) [DANILOVIC & BROWNING 2004]. As [MAURER 2007] showed the mappings allow for deducing the relations among the components due to the

context. Moreover, they allow for transforming a structure from one domain to the other. The purposes and applications of the mappings are described in the next subsections (2.3.1 to 2.3.4). Some authors propose describing several domains in one model (e.g. [EICHINGER ET AL. 2006], [BARTOLOMEI 2007] and [LINDEMANN ET AL. 2009]).

There are four important contexts of components for structural analysis. They were derived by a qualitative analysis of the literature. First, there is the development context. Components are usually specified in the concept stage of the product life cycle and are the result of planning and design processes. These processes specify components based on product requirements and functions, which are broken down to the components. Therefore, the components are closely linked to both requirements and functions. Subsection 2.3.1 describes the relations between components and requirements/functions and the reasons for modelling them.

Second, there is the parameter context. Each component has several technical parameters, which define its properties. The technical parameters may include but are not limited to dimensions and performance parameters. They are not independent but often must be consistent to each other to allow for integrating several components to working product. Subsection 2.3.2 describes the relations between the components and technical parameters and gives some example applications.

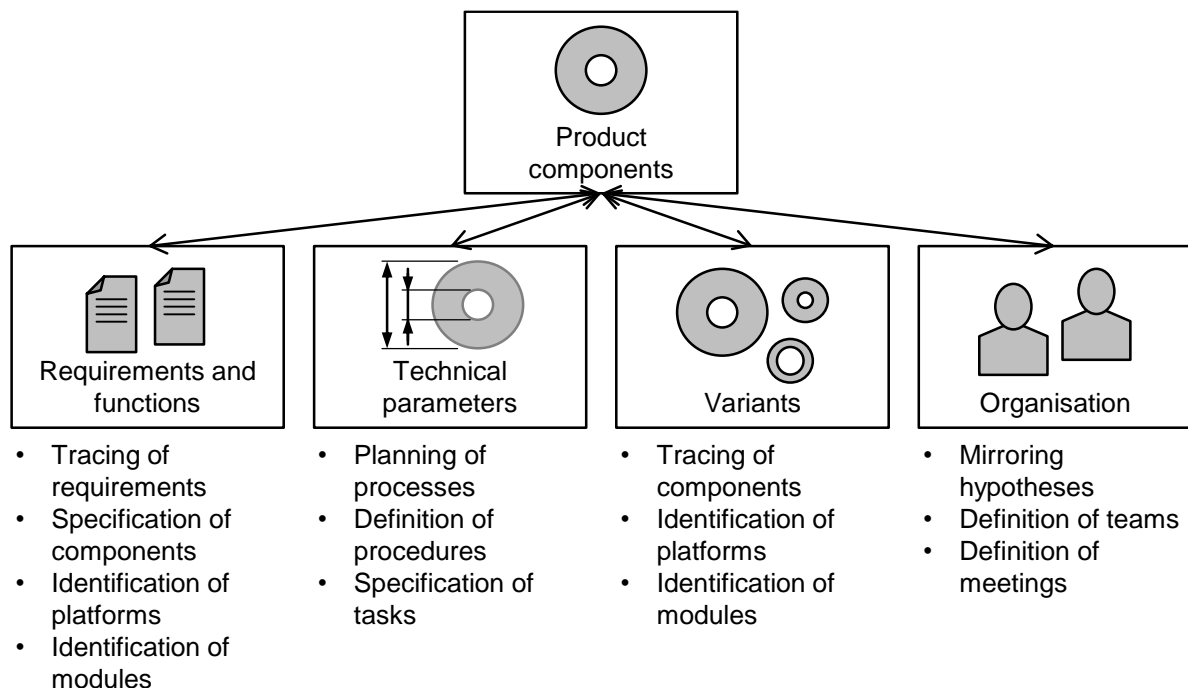


Figure 2-4 Context of network models of product components

Third, there is the variant context. Today, most components are used in several product variants as most companies operate in several markets and try to fulfil all customer needs. For cost, time and quality reasons they try to increase the rate of component reuse by applying variant strategies such as product families. Therefore, components often have to comply with the

requirement of many products or even product lines. Subsection 2.3.3 describes the relations between the components and variants and gives some example applications.

Fourth, there is the organisation context. All products and components are developed by people, who are part of an organisation. To do so the people form companies, departments and teams. Beside the assignment of responsibilities and tasks, the organisation defines the way of working and communication e.g. by meetings, workshops or virtual tools. It is commonly accepted that the organisation architecture and the product architecture should be aligned for efficient engineering of the product. Structural modelling provides tools to model both architectures, which are presented in section 2.3.4.

Several authors propose chains of domains, which mimic the creation of artefacts in the development process. Most of them describe both the design (e.g. requirements, functions, components) and the integration steps (e.g. components test and system tests) – e.g. [BRAUN & LINDEMANN 2007], [MAIER ET AL. 2007], [MOCKO ET AL. 2007], [LI & CHEN 2008], [BARTOLOMEI ET AL. 2011], [BAUER ET AL. 2011] and [GU ET AL. 2012]. Others focus on the artefact chains from development to production e.g. [GREINER ET AL. 2007] and [HELTEN ET AL. 2010]. However, those integrated model chains are not discussed in more detail.

Figure 2-4 shows the main contexts of products and gives the main reasons for modelling them. More details are given in the next subsections (2.3.1 to 2.3.4).

2.3.1 Functions and requirements

Requirements are the starting point for any component design. They define the purpose of the product and consequently the design task. The following design process first refines the requirements and transforms them later into the specification of the components, before beginning the detailed development of the components. Current research on product development recommends modelling the functions of the product as an intermediate step. In the past the requirement, function and component models were separated. Currently, there is a trend both in academia and in industry to link the models by describing the relations among them. The result is an integrated product model, which is the foundation for model-based system engineering. The relations are usually designated as fulfilment relations i.e. functions fulfil requirements and components fulfil both functions and requirements (see e.g. [KREIMEYER ET AL 2006], [BIEDERMANN ET AL. 2007] and [EBEN ET AL. 2010] for hardware and [NORD ET AL. 2011] for software).

There is an ongoing debate on the definition and differentiation of requirements and functions as well as the necessity of functional modelling. However, for the purpose of this subsection a formal definition and differentiation is not necessary as all statements apply to requirements and functions alike.

The main reasons for mapping functions and requirements to components are specification and tracing. By assigning functions to components the components' purposes are defined [ULRICH 1995]. By assigning requirements to components technical parameters, testing criteria and main dimensions as prescribed (see e.g. [BAGLEY 2005] and [MCLELLAN ET AL. 2009]). Together functions and requirements form the specification of the components [ULRICH 1995]. They

allow for dimensioning and embodiment design of the components. Through the documentation of the mapping, tracing becomes possible [DANILOVIC & BROWNING 2004]. All affected components are identified, when changing a requirement (forward tracing), and all affected requirements are identified, when changing a component (backward tracing). Forward tracing allows for effective planning and reacting to new or changed requirements. Backward tracing allows for systematically testing all functions and requirements during verification of components. Most current PLM systems have a tracing mechanism or integrate specific tracing tools [DEMOLY ET AL. 2012]. Though tracing and to a lesser degree specification use structural models and their visualisation, they make hardly any use of metric-based structural analyses.

Structural analyses are much more common for identifying platforms and modules. Both tasks use a given mapping of functions and requirements to components [MARTI 2007]. Platforms fulfil those functions and requirements, which are common to all variants of the product. Therefore, they comprise all components, which contribute to these functions and requirements. Modules are designed to simplify the development, testing and production of a product. One characteristic to achieve the simplification is encapsulating functions and requirements. Thus, modules can be developed, tested and designed separately. By analysing the mapping from functions and requirements to components sensible modules can be defined. Beside functions and requirements, variants and interfaces play an important for identifying platforms and modules. The structural modelling of variants is discussed in subsection 2.3.3; section 3.4 gives an overview over structure-based methods and procedures for identifying platforms and modules.

Though the mapping from requirements and functions to components is commonly used in structural analysis and modelling there are hardly any recommendations for the modelling process. One of the few recommendations comes from axiomatic design and states that functions and components should be refined until there a 1-to-1-mapping between them. However, this recommendation is not widely applied. Most models use an n-to-m-mapping. There are also some ideas about modelling requirements e.g. that each component must be assigned to at least one function to be well-defined. However, so far the modelling has not been conclusively discussed in literature.

2.3.2 Technical parameters

The mapping between components and technical parameters is a link between the static product model domain and the dynamic process model domain. Product models usually describe a fixed state of the model. In contrast process models describe something changing and evolving over time. Parameter networks are commonly assigned to the process domain as they model a dynamic process [KUSIAK & WANG 1995]. Some authors consider development processes as the elimination of uncertainties from the product design. They argue that each decision in development determines properties of the product, which include technical parameters.

Parameter networks usually describe the precedence relations among the parameters. Precedence means that the value of one parameter must be known, before the value of the other can be determined. Based on the precedence model a sensible order for determining the parameters can be determined. The method is variously called sequencing, triangularisation or

topological sorting. Beside the order of the parameters sequencing allows for determining groups of parameters, that can be defined in parallel as they are independent and groups that must be defined together as they are mutually dependent [KUSIAK & WANG 1995]. [BONGULIEMI ET AL. 2001] propose using parameters to check the consistency among requirement and technical solutions.

By mapping the parameter network to the components (see [MAURER 2007] for the formula) a precedence network of the components is calculated. It allows the planning of the component design process as sequencing methods can be applied to the component network as well. Moreover, it supports the definition of procedures for common design tasks, which may be automated e.g. by parametric CAD models. Finally, the definition of design tasks is supported as group of mutually depended components can be identified.

Parameter networks are usually modelled in workshops of senior designers or by interviewing engineers who are responsible for the parameters or components. [KUSIAK 1993] recommends to ask, which parameters influence the current parameter as most people know, where to get information, rather than where their results as used. Though some studies on the creation of parameter networks exist the modelling has not been conclusively discussed in literature.

2.3.3 Component variants and product variants

As stated above most companies offer a wide range of product variants to the customers. To do so the companies follow a variant strategy, which aims at covering most (at best all) costumers and limiting the costs for developing, producing and offering the variants. Most strategies rely on defining modules and platforms with standardised interfaces (see [ULRICH 1995] and [YASSINE & WISSMAN 2007]). Beside functional considerations the definition of modules and platforms often bases on an analysis of the offered and sold variants. One way to do so is structural analysis (see [BRAUN & DEUBZER 2007] and [DEUBZER ET AL. 2008]). [KESPER 2013] presents the current state of art of variant management in combination with structural analysis and introduces some new tools.

The relation between components and product variants is usually considered to be an aggregation [STEVA ET AL. 2006]. That means that the components are part of the product variant but can be used in several variants. Kusiak recommends modelling this relation a DMM, which he calls interface structure matrix (ISM) (see [KUSIAK 2007] and [KUSIAK 2008]). It lists the components on the vertical axis and the variant of the horizontal axis. This model allows for deriving platforms and modules by clustering the ISM. The platform suggestion comprises the components, which are part of all variants (i.e. having a full row in the ISM).

As the ISM is usually used for analysing and optimizing an existing variant structure it can be created very efficiently. The data for creating the model is available in several IT systems e.g. ERP, PLM and CAD systems. The simplest resource of the data is a collection of bills of materials (BOM). The ISM can be created by parsing through the bills of materials, which are available, correct and up-to-date as they are required for both purchasing and production. Though the modelling process is rather efficient and research produced some substantial results structural analysis is not commonly applied in variant management.

2.3.4 Organisations

It is generally accepted that technical products and the organisations developing and producing them are closely interrelated. [CONWAY 1968] showed that the product structure mimics the structure of the organisation developing the product (this is today known as Conway's law). Moreover, he claims that the quality of the product depends on the matching between product and organisation structure and is reduced if there are mismatches. Further research led to the formulation of the fundamental isomorphism between task and product structure: every dependency in the product must be addressed in the process; otherwise error and expensive changes will occur [BALDWIN & CLARK 2000]. It is usually recommended that the organisation and process structures should follow the product structure i.e. the organisation has to adapt to the product not the other way round.

As most of the findings and recommendations refer to dependencies and relations structural analysis is a prime candidate for handling product and organisation structures. There are two main tasks for structural analysis: deriving a suitable organisation structure from the product structure and detecting mismatches between them (see [SOSA ET AL. 2003]). Key to both tasks is a model of the mapping between the two structures (see [SOSA ET AL. 2004] and [SOSA 2007]). On the organisation side people, departments, meetings and committees may be modelled as communication networks including intensity and frequency (see [MORELLI ET AL 1995] and [GUTIERREZ 1998]). The product is usually modelled a component network. The mapping describes which part of the organisation is responsible for or contributing to the design of the component.

For deriving a suitable organisation structure from the product structure both the mapping model and a component network model are needed. The component network is then mapped into the organisation domain. The resulting organisation structure is then analysed e.g. by clustering to derive teams, committees and even departments. [BIEDERMANN & LINDEMANN 2011C] analysed the matching between the component and the communication structure of an engineering project. They showed (if not conclusively) that communication can be predicted by the component structure. They also recommended using contact relations as the basic component structure, which should be augmented by show flow relations. This applies only to hardware structures. [CASTRO 2010] provides a case study for mismatching software and communication structures.

For detecting mismatches between organisation and product structure three models are necessary: an organisation network, a mapping model and a component network [SOSA 2008]. Then, the component network is mapped to organisation domain or vice versa [SOSA ET AL 2003]. The resulting network is then compared to the native network e.g. by computing a delta DSM (see [SMALING 2005] and [HOFSTETTER & DE WECK 2007]). The comparison is usually done in the organisation domain as the detected mismatches allow for improving the organisation [DIAZ-GARCIA 2009]. To comply with the fundamental isomorphism new communication paths may be introduced to avoid error; others may be pruned to make the organisation more efficient. [MAURER & LINDEMANN 2007] describes a case study, which shows the potential of combined analysis of the product and organisation structure.

2.4 Conclusion

This chapter first discussed structural component models of hardware and software. The models of the context of product components were introduced. Though the breakdown structure (i.e. the hierarchy) of components was described the focus lay on the interface structure. Moreover, the main differences between hardware and software concerning their structural models were highlighted.

The purpose of this chapter is to determine, which structural models are most commonly used to analyse products and their components. Thereby, the most important models and analyses are researched in the remaining thesis. Thus, the significance of this thesis is ensured as it addresses commonly applied tools and methods.

This thesis focuses on hardware models as they are more commonly structurally analysed. From the analysis of the state of the art two main conclusions for the remaining thesis were drawn: if not stated otherwise the structural models describe a network at the component level (according to the definition in Table 2-1) with contact as type of relation (according to the definition in Table 2-2). Contact is the most common type of relation in the structural modelling of hardware components. The choice of the level of abstraction is arbitrary as there is no conclusive research on choosing the level of abstraction.

Though the thesis mainly addresses hardware models, the next chapter also introduces analyses for software models since the corresponding research uses rigorous methods and addresses similar questions. Thereby, additional research methods and opportunities are presented, which mainly result from the differences in models creation for hardware and software. However, the differences of the nature of hardware and software models lead to the open research question if results derived from hardware can be transferred to software or vice versa.

The discussion of the state of the art raised some open points for research, which are not approached in this thesis. All of them deal with choosing and creating appropriate models for structural analysis: choosing the level of abstraction, selecting the types of relations and choosing the data acquisition method. So far only preliminary results have been published.

3. Structural analysis of component networks

The previous chapter introduced models for describing product structures and derived the main types of models relevant for structural analysis. This chapter presents the methods and purposes of structural analysis, once a structural model is available.

As discussed before this thesis focuses on the structural metrics. As most metrics are closely linked to or derived from other analyses the next section (3.1) gives a more general introduction to structural analysis. It highlights in particular the computation and algorithms of the analyses and presents a classification scheme. As all metrics can be traced back to graph theory, the section also gives a short introduction to this field of mathematics.

The remaining sections (3.2 to 3.5) present applications of structural analysis of product structures. The focus lies on structural metrics applied to models of hardware components and contact relations as this is the most common structural model (see section 2.4). The list of applications results from a qualitative analysis of literature. The most common applications of metrics are the assessment of life-cycle properties and the assessment of product-inherent properties.

Life-cycle properties like changeability of a product (also known as non-functional requirements and “ilities”) do not manifest themselves primarily in engineering projects but rather in the whole life-cycle. They mostly refer to modifications of a product due to new requirements, new limitations and new applications during design, production or usage of the product. The assessment of these properties during the concept phase is important as they emerge from the product design but are hardly measurable by conventional quality methods. Section 3.2 gives definitions of the most important life-cycle properties and presents structure-based methods for assessing them.

Product-inherent properties like modularity emerge from the interactions among the components. Unlike the life-cycle properties they only depend on the product design and not the product life-cycle. They are particularly important for the engineering of the product as they support (modularity) or limit (complexity) the effectiveness of the development phase. Section 3.3 gives definitions of the most important product-inherent properties and presents structure-based methods for assessing them.

The remaining two sections introduce applications of structural analysis, which usually rely on component models but do not use metrics extensively: product decomposition (section 3.4) and change propagation (section 3.5). Product decomposition is the activity of breaking a product down into modules and components to allow for efficient development and production. It is closely linked to product modularisation, which is the activity of grouping components into modules with the same intention as decomposition. Both of them usually rely on visualisation rather than metrics and make use of matrix reordering and clustering. Change propagation is the phenomenon that changes in a product may cause further changes due to the interaction among components. Change propagation is commonly researched with simulation approaches. However, there are some publications, which discuss the prospects of structural analysis in this

field. Figure 3-1 gives an overview of the main purposes and outcomes of structural analysis and presents the structure of this chapter.

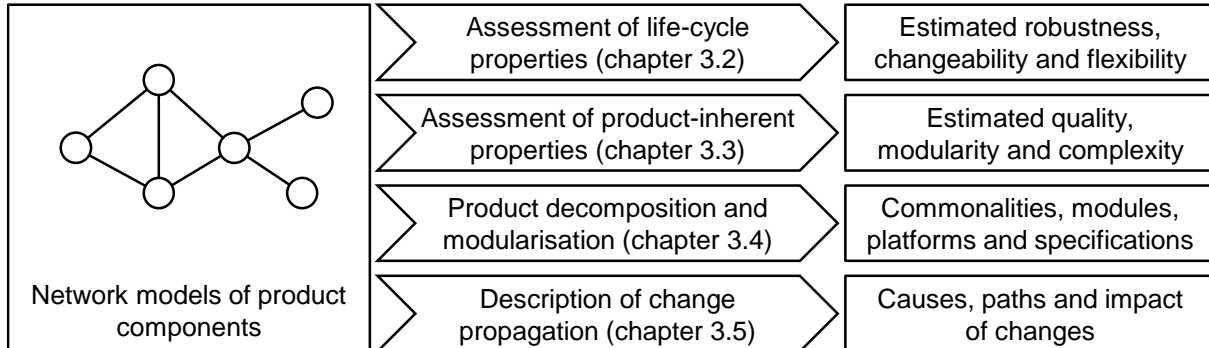


Figure 3-1 Purposes and outcomes of structural analysis of product networks

3.1 Fundamentals of structural analysis

Structural analysis has several roots: engineering, management science, computer science and mathematics. Engineering and management science developed their own forms of structural analysis (e.g. sequencing of DSMs or portfolio diagrams) as they needed a tool to deal with the structure of products and other complex systems. Most of these engineering-based analyses were created intuitively using tools already available (this is why matrix representations are still more common than network representations). Only in the past ten years structural analysis in engineering and management science made use of the results from computer science and mathematics. Structural analysis bases on the mathematical field of graph theory, which deals with networks of nodes connected by edges. An important subfield is algorithmic graph theory, which provides the algorithms for handling structural models. In combination with computer science graph theory provides a wide range of structural analyses, which can be computed efficiently. Recent advances e.g. motif analysis have spawned an almost infinite number of analysis tools. Subsection 3.1.1 presents the most important results from graph theory for structural analysis. Subsection 3.1.2 classifies the available analyses, which allows for limiting the scope of this thesis.

3.1.1 Graph theory, network theory and matrices

Graph theory is the branch of mathematics dealing with graphs. “A graph is a representation of a set of objects where some pairs of the objects are connected by links” [HARARY ET AL. 1965]. This basic definition has been refined several times to differentiate between fundamental types of graphs: undirected, directed and mixed graphs; simple graphs, quivers and multigraphs. In this thesis undirected, simple graphs are used. That means that the links between the objects have no direction and there are no links connecting one object to it itself. This is the simplest definition of a graph in graph theory. More refined classes of graphs e.g. regular graphs have been introduced but are not relevant for this thesis.

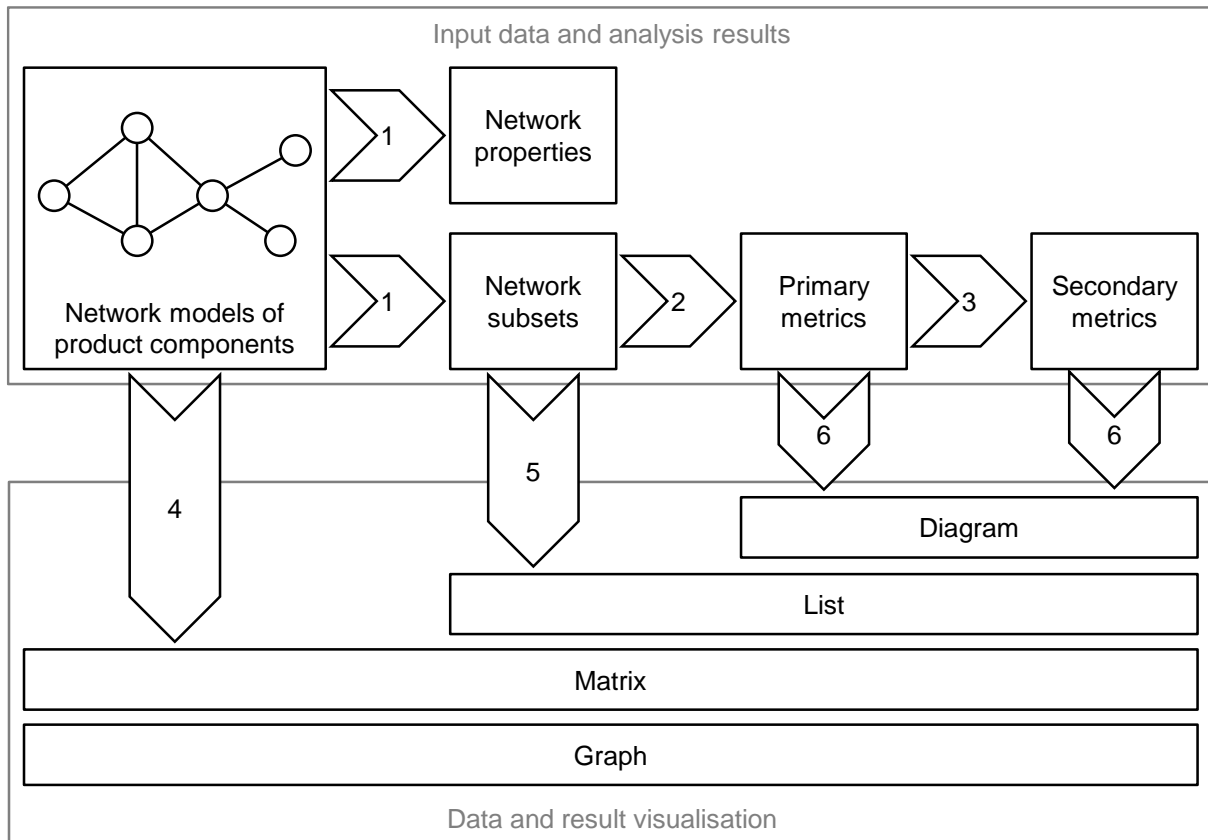
In graph theory there are three major representations of graphs: set notation, network and adjacency matrix. The set notation is very formal and mainly used in definitions, propositions, lemmas and proofs. The network representation is usually used for showing particular graphs as it is intuitively comprehensible to most people. The adjacency matrix is a well-known representation but hardly used outside engineering. All three representations show exactly the same data and can easily be transformed into each other.

Graph theorists defined several properties and subsets of graphs, which have been applied in engineering. For example the results on planar graphs (i.e. the graph can be drawn without two links crossing each other) are used when designing circuit boards. However, the properties of graphs have hardly been applied in the field of structural analysis. The subsets of graphs play a more important role in particular in this thesis as they are the base for most structural metrics. The most important types of subsets are paths, cycles, blocks, cliques, and trees (see appendix 11.1 for a definition). Each of these types has been introduced in graph theory due to applications in other fields of science. For example, the theory of trees was established by Gustav Robert Kirchhoff when he researched electrical circuits [HARARY ET AL. 1965].

The major metric in graph theory is the degree (i.e. the number of links a node has; see appendix 11.3 for a formal definition). It is the basic property to describe the connectedness of a node as there has been extensive research on its value and its distribution. Other metrics have been proposed but are not as widely used as the degree. The most important metrics are distance, order and size (see appendix 11.2 and 11.3 for a definition). Research on structures in engineering and sociology has proposed several additional metrics, which are discussed in the next sections.

A particularly prolific field of graph theory is algorithmic graph theory. It researches algorithms for the efficient computation of graph properties, subset and visualisations (see [BATTISTA ET AL. 1998] for visualisation and [GROSS & YELLEN 2003] for a general introduction; both give an excellent overview on the algorithmic side of graph theory). Though there are no efficient algorithm known for some problems of graph theory computation of them is not an issue as very efficient heuristics and approximation are available. One of the most important results was the advent of motif analysis, which allows for almost any analysis thinkable [MILO ET AL. 2002]. The algorithms of motif analysis can find arbitrary subsets in a graph. Thereby, an almost infinite number of metrics becomes available.

A comparatively recent field of graph theory is network theory. It does not originate from mathematics but rather from the analysis of natural networks. Network theory is considered to be part of the nonlinear sciences, which is an interdisciplinary field of research usually associated with physics. The main hypothesis is that natural networks have the same properties and follow the same rules of evolution. The hypothesis results from the observation of several natural networks: citation networks in academia, linking networks of the internet, molecular networks of proteins and so on. There have also been some publications with an engineering background. Network theory introduced three major characteristics for the analysis of structures: degree distribution, clustering coefficients (see also appendix 11.3) and average path length (see also appendix 11.2). Together they allow for classifying networks; with random and small-world networks being the most important classes. [CAMI & DEO 2008]



- 1 Computation by graph-theoretic algorithms
- 2 Computation by frequency analysis of subsets
- 3 Computation by statistical metrics or combination of primary metrics
- 4 Visualisation of whole networks as graph or matrix
- 5 Visualisation of single subsets as graph or matrix, of several subsets highlighted in whole network or as entries in a list
- 6 Visualisation of metrics as a diagram, as a list or as coloring of a whole network

Figure 3-2 Classification of structural analyses

3.1.2 Classification of structural analyses

The previous subsection gives a short introduction to the theoretical background of structural analysis. It showed that there is an almost infinite amount of analyses available due to advances in graph theory. In this subsection the structural analyses are classified and the major types are defined. Finally the scope of this thesis in terms of structural analysis is given.

Figure 3-2 shows the main types of analysis and their interrelation. The field of structural analysis comprises two main parts: the data part and the visualisation part. The data part is shown in the upper section of the figure. It depicts the complete computation process starting from the networks to the primary and secondary metrics. The both computationally and algorithmically most challenging part is the computation of the network subsets. The

computation of the metrics is rather simple and fast. Table 3-1 gives short definitions for each type of analysis.

The visualisation part is rather more straightforward. Matrix and graph visualisations represent whole networks ([BATTISTA ET AL. 1998] and [HARARY ET AL. 1965]). Both of them allow for showing additional data e.g. by colouring the nodes in a graph according to the values of a metric. Lists give all details on subsets and metrics but may be rather confusing when dealing with huge networks. Finally, diagrams are particularly important when visualizing metrics clearly and compactly [KREIMEYER & LINDEMANN 2011].

This thesis focuses on the metrics in structural analysis as discussed in section 1.3. Moreover, it does not deal with visualisation. The set of research metrics will be introduced in the methodology part of this thesis (chapter 5). Beside the collection of the metrics (see section 5.2) the dependencies among structural metrics will be introduced (section 5.1).

Table 3-1 Structural analyses and visualisations

Analysis	Definition	Example
Network property	Boolean property of a whole network	Planarity
Network subset	Part of a network complying to a definition	Path
Primary metric	Metric derived from a single subset or a collection of subsets	Path length
Secondary metric	Metric derived from a primary metric or its distribution	Average path length
Graph	Visualisation of a network as graph (“boxes and arrows”)	Force-directed graphs
Matrix	Visualisation of network as a table (“rows and columns”)	Clustered DSM
List	Visualisation of a network or its subsets as a listing	List of shortest paths
Diagram	Visualisation of a network or its metrics as a plot	Influence portfolio

3.2 Assessment of life-cycle properties

This is one of the most common applications of structural analysis. The aim is to identify system elements, which are important to the system behaviour during its life-cycle. This improves the handling of the system and leads to optimised systems. Most life-cycle properties are associated with modifications of the product during its development, production and usage. They differ in terms of various dimensions: internal or external triggers, effectiveness vs. efficiency, modification of parameters, components or structures, and so on.

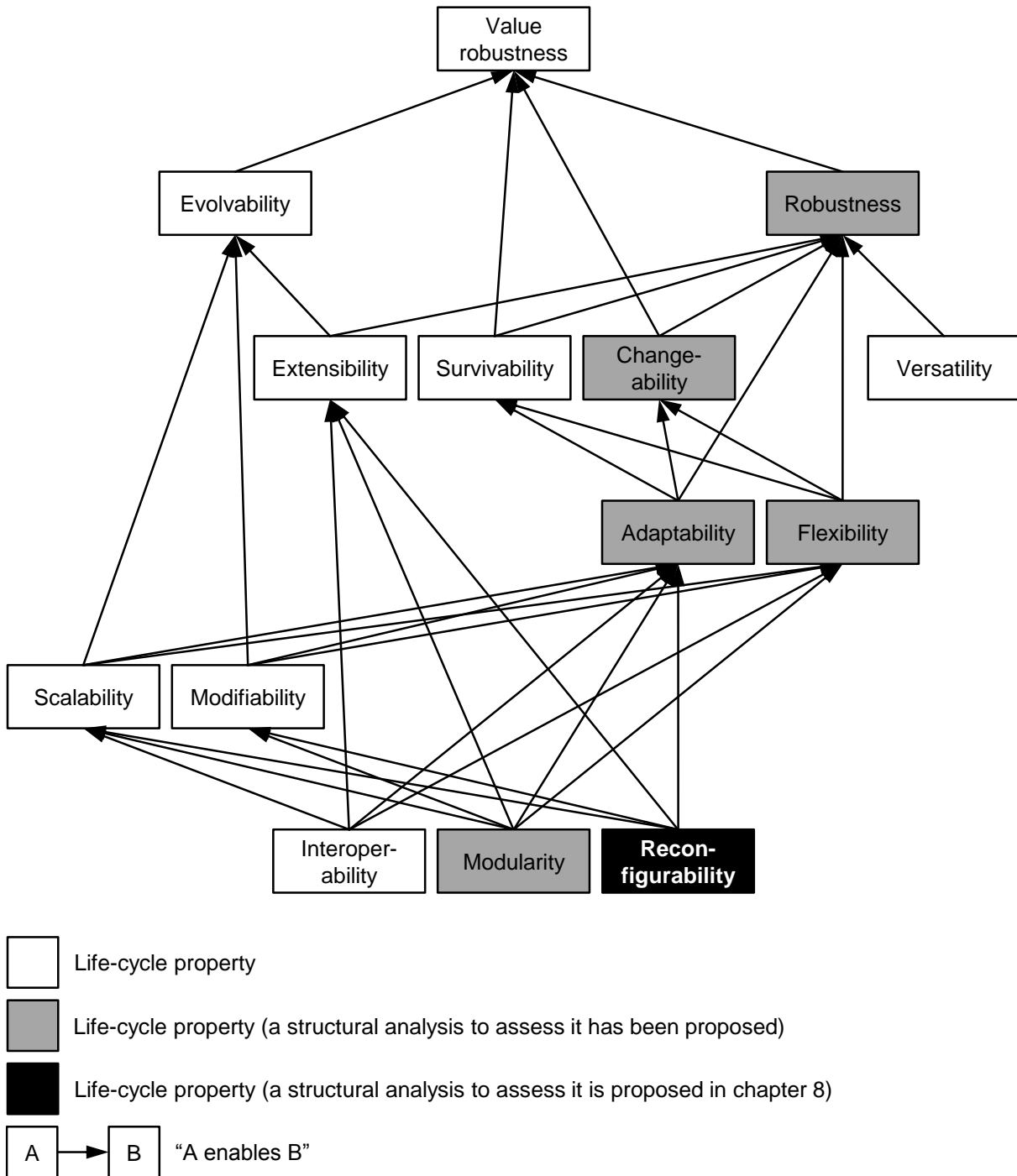


Figure 3-3 Enabler network of life-cycle properties based on [DE WECK ET AL. 2012]

Life-cycle properties depend on each other. There is a wide body of research on life-cycle properties. One set of publications deals with designing products with some particular life-cycle properties. The publications belong to the design for X community where X stands for a particular product property. Other publications deal with the assessment and measurement of life-cycle properties both during the entire life-cycle and during the concepts phase. However,

many of these publications state that one property is needed to achieve another one. [DE WECK ET AL. 2012] did a meta-analysis of the enabler relations. The resulting network is shown in Figure 3-3. Modularity is considered as a product-inherent property in this thesis and discussed in the next chapter. Table 3-2 gives the definitions of the properties.

Structural metrics are used to rate the elements by estimating the life-cycle property. This means they do not allow for measuring the properties directly. So far structural analyses have only been discussed for a few life cycle-properties. These properties are highlighted in grey in Figure 3-3: robustness, changeability, adaptability, flexibility and modularity. The first four properties are discussed in the next subsections (3.2.1 to 3.2.4); modularity is discussed in subsection 3.3.2 as it is a product-inherent property. Finally, this thesis proposes structural metrics for assessing reconfigurability in chapter 8.

Table 3-2 Life-cycle properties according to [DE WECK ET AL. 2012]

Property	Definition
adaptability	to be changed by a system-internal change agent with intent
agility	to change in a timely fashion
changeability	to alter its operations or form, and consequently possibly its function, at an acceptable level of resources
evolvability	design to be inherited and changed across generations (over time)
extensibility	to accommodate new features after design
flexibility	to be changed by a system-external change agent with intent
interoperability	to effectively interact with other systems
modifiability	to change the current set of specified system parameters
reconfigurability	to change its component arrangement and links reversibly
robustness	to maintain its level and/or set of specified parameters in the context of changing system external and internal forces
scalability	to change the current level of a specified system parameter
survivability	to minimise the impact of a finite duration disturbance on value delivery
value robustness	to maintain value delivery in spite of changes in needs or context
versatility	to satisfy diverse needs for the system without having to change form (measure of latent value)

3.2.1 Robustness

Robustness is one of the main target “ilities” to be achieved in a product. A product is robust if the product maintains its specified functionality and parameters under a changing environment. Though, there has been some research on robustness it is still not known if the architecture determines the product’s robustness or if it results from the component design [CRAWLEY ET AL. 2004].

There are some publications dealing with structural robustness. [BHUSHAN 2005] presents a method for robust inventive software design, which combines DSMs, TRIZ and AHP (analytical hierarchy process). However, the robustness evaluation does not use structural methods but results from the TRIZ methods. [KISSEL ET AL. 2011] propose two metrics for evaluating the structural robustness of a product concept: the impact of an element when changed and the probability of an element to change. The impact is a weighted sum of in- and out-degree of the neighbouring elements. The probability is a normalised out-degree. They also provide some evidence that decoupled and/or modular concepts are more robust.

Though there is some evidence that the product structure impacts the product’s robustness there are no empirical long-term study how strong or valuable that impact is.

3.2.2 Changeability

Changeability is one of the main target “ilities” to be achieved in a product. A product is changeable if its operation, form and function can be altered at an acceptable level of effort. Changeability is connected to several other “ilities” and design concepts: modularity (see subsection 3.3.2), modularisation (see section 3.4) and change propagation (see section 3.5). Structural modelling and analysis has four purposes when dealing with changeability: describing the change, estimating the change impact, estimation the change likelihood and measuring the changeability of a concept.

Several methods for describing the changes of product structure have been introduced in chapter 2 e. g. [EBEN ET AL. 2008] and [DE WECK 2007]. These methods are mainly for modelling rather than analysing changes. [LAMANTIA 2006] introduces the change ratio as a metric for measuring the extent of changes between two software product generations. It is the ratio of the number of added and removed components and the total number of components. Though many publications propose estimations for the extent of potential changes (see next paragraph) most of the metrics have not been applied to real changes.

Estimations of the change impact and the change likelihood are basic data for analysing and predicting change propagation. Most commonly both estimations are not derived from the structural model but are input data for it. In this case the estimations are derived by interviewing engineers and managers (see e.g. [KOH ET AL. 2007]). [MARTIN & ISHII 2002] propose the generational variety index, which combines both estimations. They use the index for measuring future design effort. The impact and likelihood estimations are also input for the product variant portfolio and the propagation absorber/multiplier portfolio introduced by [KOH ET AL. 2007]. Both portfolios allow for choosing an appropriate design strategy for the product components depending on their position in the portfolios. Another common approach for estimating the

change impact and the change likelihood is the creation and valuation of change scenarios (see e.g. [SULLIVAN ET AL. 2001] or [CAI & HUYNH 2007]). Usually, the scenarios are combined with net option values considerations and modularity approaches for evaluating the latter. Most of the work on estimating change impact and likelihood is based on simulative and theoretical research. One of the few empirical works by [PASQUAL 2010] shows that structurally central components are correlated with change propagation. It also shows that centrality metrics are not sufficient for finding high impact components.

Finally, [ROSS ET AL. 2008] propose an approach for measuring changeability. They do not model the coupling among components but a network of parameter states, which is enhanced by cost estimations for each state and each transition between states. The changeability of state is quantified by the filtered out-degree i.e. the number of neighbouring states, which do not exceed a cost threshold.

Many approaches use structural models for dealing with or measuring changeability or some of its aspects. Most of them do not rely on purely structural models but require additional data such as cost, impact or likelihood estimations. Moreover, there is hardly any empirical work testing the claims of the theoretical and simulative results.

3.2.3 Flexibility

Flexibility is one of the main target “ilities” to be achieved in a product. A product is flexible if it can be intentionally changed by an external agent. Structural analysis has two main purposes when dealing with flexibility: guiding the engineers when creating flexible designs and assessing the degree, to which a system or component is flexible. Flexibility is also a feature of the organisation. Changes across organisational boundaries tend to limit flexibility [DALEIDEN 1999].

Most approaches for guiding engineers aim at identifying components, which are likely to be frequently changed or are costly to change. Similar to the approaches concerning changeability (see subsection 3.2.2) change scenarios are derived by interviewing engineers and managers. Financial options theory allows for assessing the value of flexible designs in these scenarios [CRAWLEY ET AL. 2004]. Other approaches reuse models from change propagation analysis. For example [SUH ET AL. 2007] propose the change propagation index for identifying candidate components for embedding flexibility. The change propagation index is the difference between the out-degree and the in-degree of the physical change propagation network among the components. [WILDS 2008] combines change scenarios and change propagation analyses. She introduces the desired flexibility score, which combines cost of change in a scenario with the likelihood of change propagations. Thereby potential components for embedding flexibility are identified.

[TILSTRA ET AL. 2009] present an approach for assessing the flexibility of a product. They link design for flexibility guidelines to structural models of the component interactions. By this combination they determine if the product design fulfils the guidelines. The assessment is based on structural metrics such as the number of relation of a specific type. The individual metrics depend on the individual guidelines and vary in terms of type of relation. However, they only present a demonstrating case study, which compares two products.

Most of the structural approaches for dealing with flexibility aim at finding components where flexibility is needed or easily embedded. These publications are often closely linked to contributions to changeability and change propagation. Hardly any contribution aims at assessing the degree of flexibility itself. Moreover, there is hardly any empirical work testing the claims of the contributions.

3.2.4 Adaptability

Adaptability is one of the main target “ilities” to be achieved in a product. A product is adaptable if it can be intentionally changed by an internal agent. Adaptability is closely linked to modularity (see subsection 3.3.2) and modularisation (see section 3.4) as it is claimed that modular products are easier to adapt (see e.g. [ARTS ET AL. 2008]). Structural analysis is hardly applied in research of adaptability. [CRAWLEY ET AL. 2004] claim that adaptability depends on the design of the internal interfaces, the sensors, the control algorithms and the human interface. All of these are non-structural properties of the product. [ARTS ET AL. 2008] present a structure-based modularisation method for adaptable products. Though the result is structural the method uses action plans as additional input.

3.3 Assessment of product-inherent properties

This is one of the most common applications of structural analysis. The aim is to choose good product concepts and to find possibilities to improve them. Many product-inherent properties cannot be assigned to or broken down to individual components like weight but result from the interaction of all components like power consumption. Research in structural analysis on these emergent properties focused on quality, modularity and complexity.

Product-inherent properties depend on each other. In particular quality, modularity and complexity are closely linked to each other as shown in Figure 3-4. Table 3-3 gives the definitions of the properties.

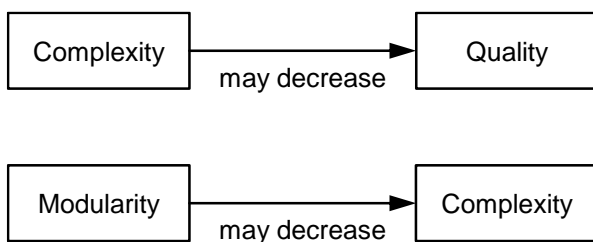


Figure 3-4 Classification of product-inherent properties

Quality is together with time and cost one of the three major aims in product design and production [LINDEMANN ET AL. 2009]. Therefore, it is often discussed in various contexts. One often stated claim is that high design and organisational complexity lead to low product quality. Many publications claim that modular products design are less complex and should be

preferred. For all three properties structural analyses have been proposed as analysis tool. These mostly metric-based analyses are introduced in the next subsections (3.3.1 to 3.3.3).

Table 3-3 Product-inherent properties

Property	Definition
modularity	degree to which a product is composed of modules
complexity	degree to which a product is comprehensible/predictable in its behavior
quality	degree to which a product is fit for its purpose

3.3.1 Quality

Quality is one of the main emergent properties to be achieved in a product. It is the degree to which a product is fit for its purpose. Quality is closely linked to complexity as many authors claim that design or organisation complexity reduce the quality of a product. There are two main structural approaches to deal with quality issues: the organisational approach and the interaction approach. The organisational approach focuses on the context of the product components to check if high quality can be achieved by the design set-up. The interaction approach focuses on the relations and the interaction among the components. Most publications on the interaction approach deal with software products whereas the organisation approach makes no specific statements which explicitly differentiate between hardware and software. Yet, it is mostly discussed for hardware products.

The organisational approach relies on context models of the product components (see section 2.3 for an introduction). [MAIER ET AL. 2007] introduce a modelling scheme, which links requirements to tests via a series of seven domains including functions, components and parameters. By computing indirect dependencies the coverage of requirements by tests can be determined and allows for identifying missing tests. [CHENG 2007] uses a similar yet, independent multi-level approach for comparing weighted requirements and the testing strategy. [KORTLER ET AL. 2009] focus on the link between requirements and functions and use degree metrics for guiding the refinement and specification processes of the two domains. Contrary to these artefact-based approaches [GOKPINAR ET AL. 2010] discuss the role of the alignment between product and organisation architecture. They show that misalignment leads to an increase of quality issues during later design and life-cycle stages. Though all contributions to the organisational approach provide arguments for the validity of their results and claims they mostly do not provide empirical or experimental evidence.

The interaction approach relies on models of the relations among components in particular among software components (see subsection 3.3.2 for an introduction). [SOSA ET AL. 2008] analyse the structure of 20 software application over seven generations on average and its influence of software quality namely the number of detected and resolved bugs. They use seven

metrics to analyse the software structure including the number of cycles per node (both total number and filtered by module) and the distance centrality. The three main results are: many cycles per node increase the number of bugs, many cycles across module boundaries decrease the rate of bug resolution and high distance centrality increases the need for improvement. [AKAININE 2010] uses a similar research approach but focuses on maintenance of software. He compares two software generations to show the impact of a fundamental redesign. The maintenance effort results from an analysis of the maintenance activities for thirty months after each release. The software structure is mainly analysed by active and passive reachability of each component. The metrics are also used for components classification in a portfolio. The analysis results show the change of the software structure (much less core components) and the impact of the structure on maintenance (less structural complexity leads to less effort, less rework and higher productivity). Together the results prove the success of the software redesign. [SOSA ET AL. 2011] analyse the impact of the degree distribution onto software quality namely the number of software defects. They partially use the data from [SOSA ET AL. 2008]. The results show the existence of hubs, i.e. components with high degree values, reduces the number of software defects. The results were tested against seventeen control variables to show that the reduction effect really results from the software structure. These three contributions to the interaction approach propose tools for predicting potential quality problems and prove the impact of the software structure on product quality. More importantly, all three contributions base on rigorous empirical analysis.

The contributions to structural analyses of product quality fall into categories. The organisational approach uses context models, provides arguments for validity and addresses products in general. The interaction approach uses relations among components, bases on empirical analysis and focuses on software products. The main reason for this difference is the availability of data for research. As discussed in section 2.2 software structures can be easily derived from the source code. Therefore, many more products and product generations can be analysed. Moreover, quality assurance is much better documented for software in particular for open source software. Therefore, the data base for research is very big. If the results for software can be transferred to hardware is still an open research question (see sections 2.1 and 2.2 for a discussion).

3.3.2 Modularity

Modularity is one of the main emergent properties to be achieved in a product. It is the degree to, which a product is composed of modules. Modular designs are often claimed to be easier to design, test, produce and maintain in particular when dealing with complex environments, organisations and technologies. However, under which conditions modular designs are superior to integral designs is still an open debate. [GERSHENSON ET AL. 2004] provide an overview over modularity measures and design methods. Their most important insights are: the measures are a least to a certain degree subjective; and, “measures and methods lack rigorous verification and validation”. [SHARMAN ET AL. 2002B] define a vocabulary for modular and integrated products. Contrary to [GERSHENSON ET AL. 2004] this section purely focuses on the assessment of modularity using structural models and/or analyses. Modularity is closely linked to product decomposition and modularisation, which is discussed in section 3.4. There are three main

approaches: structural modelling but not analysis; structural analysis of entire networks and structural analysis at component level.

A typical tool for assessing modular designs (see also section 3.4) is the combination of design structure matrices for modelling and real options theory. [SULLIVAN ET AL. 2001] use this combination to show that modular designs are superior for software products. However, they do not measure the modularity of the design but refer to design proposals by [PARNAS 1972]. Later approaches combine structural metrics with other key figures. [LARSES 2005] proposes a balanced scorecard for assessing the modules of automotive architecture. Beside two structural metrics (average degree and relational density – both relative to each module) he uses key figures for reliability, cost and weight and provides a case study. [BADOR ET AL. 2007] use a set of metrics to analyse the commonality within the cockpits of aircraft families. The set of metrics uses hardly any structural models but mostly uses additional data such as costs and take rates. [CAI & HUYNH 2007] use a changeability metric for measuring modularity. All these approaches are very data intensive and require quite a lot of data acquisition or estimation, which goes well in line with the observations by [GERSHENSON ET AL. 2004].

Structural analysis of an entire component network assesses the modularity of the whole product concept. [SHARMAN ET AL. 2002B] use the visibility-dependence signature (i.e. a portfolio of active and passive reachability) to characterise modular and integral products. They provide reference portfolios for various types of products but also show in their case study that real products are a mix of various types. [MACCORMACK ET AL. 2006] introduce a set of three structural modularity metrics: relational density, density of the reachability matrix and the so-called dependency cost. The latter is a weighted form of the reachability metric; the weights differ depending on the position of the relation in the matrix. In their case study they compare several releases of two large open source software packages. The results show the increase of modularity due to redesign projects and optimised design organisation. [LAMANTIA 2006] uses the same metrics to show the benefits of modularity in software. His two case studies show that more modular products allow for experimentation in and substitution of modules without redesign the entire software. [HÖLTTA-OTTO & DE WECK 2007] propose two metrics for measuring modularity: relational density and the singular value modularity index. The latter is an estimation of the decay of the singular values of a DSM. They make three studies to evaluate the metrics. Idealised reference structures for integral, bus-modular and modular products determine the extreme values for the metrics. A comparison of real product structures and random structures with the same relational density shows that real products significantly differ from random structure. Finally, a comparison of product pairs with the same functionality but different modularity (e.g. cell phone vs. desk phone) shows that the metrics allow for assessing the modularity of products. [CHIRIAC ET AL 2011B] use three metrics to assess the modularity of a product structure: the relational density outside module, the difference between the relational densities inside and outside modules and the minimal description length. The latter stems from information theory but does not provide additional insights. They show the applicability of the metrics with four sets of idealised structures, which differ in number of relations, number of modules, sizes of modules and level of granularity. Together with a case study the results show that the level of granularity of affects the analysis results. For example, the structure is integral at a higher level but modular at a more detailed level in the case study. However, due to a lack of data this cannot be generalised.

Structural analysis at component level assesses the modularity of single components within a product. [SHARMAN ET AL. 2002B] propose active and passive reachability as metrics for components. The lower values indicate a higher the degree of product decomposition. They also point out that the values depend on the general product structure e.g. a layered structure shows only few distinct values. [AKAIKINE 2010] uses the reachability metrics to classify software components into peripheral, shared, control and core components. [SOSA ET AL. 2003] use a variation of the degree metric to assess the modularity of subsystems in a product. A subsystem is modular if it is connected to few subsystems and integral if it is connected to many or all subsystems. Based on this classification they analyse the impact of the type of subsystem on design team interactions. They find that modular subsystem correlate with unpredicted communication and integrative teams can overcome organisational boundaries more easily. [SOSA ET AL. 2007A] propose using metrics from social network analysis to analyse the modularity of product components. They modify three metrics proposed by [FREEMAN 1978]: degree, distance centrality and path centrality. Their case study shows that the metrics allow for highlighting the role of modularity for component redesign.

Though the contributions to modularity assessment by structural analysis vary in approach (single metrics vs. sets of metrics), model (purely structural vs. extended models), focus (subsystem vs. component level) and scope (entire product vs. single components) they share one common theme: no approach measures modularity directly. They rather assess what distinguishes products, which are accepted to be modular from products, which are accepted to be integral. Some contributions use idealised structures e.g. [HÖLTTA-OTTO & DE WECK 2007] and [CHIRIAC ET AL 2011B]. Others compare the results of restructuring efforts to the initial system e.g. [MACCORMACK ET AL. 2006] and [LAMANTIA 2006]. Finally, some contributions accept the metrics as they are and simply highlight how they impact design processes e.g. [SOSA ET AL. 2003].

Moreover, the contributions to modularity measurement show two at least questionable tendencies in structural analysis research: definition of new analysis tools and development of new computation methods. For example, [SOSA ET AL. 2007A] modify metrics proposed by [FREEMAN 1978] without testing the original metrics for applicability. In this case the modifications lead to mathematical problems (namely division by zero), which do not exist in the original metrics. [MACCORMACK ET AL. 2006] describe a way to compute the reachability matrix based on matrix multiplications. However, more efficient and easier to implement algorithms have been proposed by graph theorist (see [DEO & PANG 1984] for an overview of algorithms).

3.3.3 Complexity

Complexity is an emergent property, which is often considered detrimental in a product. This thesis uses following definition: degree to, which a product is comprehensible/predictable in its behaviour. Other definitions (e.g. number of systems states or variety of elements) have been proposed but no definition gained wide-spread acceptance. Moreover, complexity is not clearly understood: some consider it to be a system-inherent property; others consider it to be a feature of human perception. Though there is no consensus what complexity actually is, there is a consensus about the implications of complexity for product design and development. Complex

products tend to be more error-prone and therefore face more changes during development, require more tests during verification and have more quality problems in production and maintenance. On the other hand complex designs are harder to copy and offer more or better functionality. To a certain degree complexity shows the contrary properties to modularity. The structure of a product is commonly accepted to impact the product complexity. Therefore, several structural metrics for product complexity have been proposed. Some of them claim to measure complexity absolutely; most of them only claim to allow for comparing product designs. [LINDEMANN ET AL. 2009]

Usually several metrics are proposed to deal with different facets of complexity. [HOLMQVIST & PERSSON 2003] propose six dimensions of complexity: number of parts, number of variants, number of solutions per function, number of technologies, size variation and modularity. The first four metrics can be derived from structural models. Size variation requires additional data and modularity metrics are discussed in subsection 3.3.2. Based on these dimensions they show that modularisation methods are less useful for complex products. [SOSA ET AL. 2007B] use complexity metrics to assess the dynamics of software architecture. They call the metrics intrinsic complexity, average module-internal complexity and cross-module complexity. Intrinsic complexity is the product of the number of elements and the number of relations. Module-internal complexity is the intrinsic complexity of a module. Both metrics exist in three forms: total, feed-forward and feedback. Cross-module complexity is the sum of the relations between modules. Sosa et al. show that higher complexity leads to higher effort in redesign and changes. [AMERI ET AL. 2009] propose two metrics for measuring the size and coupling complexity in engineering design. The size metric measures the entropy of the design and combines the number of design variables, the number of constraints, the number of modules and the number relations in the product. The coupling metric is the number of relations with have to be removed to separate the structure. Using three products Ameri et al. show that both metrics are independent. They also show that the metric values depend on the product representation. Although they discuss several complexity measurement approaches they do not compare them in their case study. [MATTHIESON & SUMMERS 2009] extend the work by Ameri et al. and introduce two additional metrics: degree of freedom and all-pairs shortest path. The first metric is the sum of all degrees of freedom (which are an additional input). The second is the sum of all distances among the components. [SUMMERS & SHAH 2010] wrap up the works by Ameri et al. and Matthieson & Summers and extend them by (non-structural) metrics for the solvability of design problems. [KORTLER ET AL. 2009] propose a complexity metric based on the planarity of the product structure. The metric is the minimal number of removed relation to achieve a planar structure (i.e. a structure without crossing relations in a network representation). Though they give an overview of structural metrics and argue that planar structures are easier to comprehend, they do not present a real case study or other empirical evidence. [HOSSAIN & ZULKARNINE 2011] combine social network metrics and modularity metrics to assess the complexity of software. They compare two software packages using five metrics: average distance, average clustering coefficient, average degree, number of strongly connected components and density of the reachability matrix. Though the metric values differ Hossain & Zulkarnine do not draw further conclusions. [MARTI 2007] proposes a metric for measuring the contribution of component to the physical complexity and the functionality of a product. The complexity metric is a weighted sum of the number parts, number of variants,

number of interfaces and the number of interface variants. Based on the metrics Marti proposes guidelines of action. The entire approach is demonstrated, tested and evaluated in four case studies.

Similar to the modularity metrics (subsection 3.3.2) no approach measures complexity directly as no quantifiable definition of complexity has been proposed so far. Apart from [MARTI 2007] no contribution assesses the contribution of individual components to the entire product. Though some contributions discuss several complexity metrics (e.g. [AMERI ET AL. 2009]) there is no comparison of the newly introduced metrics with the already proposed metrics based on the available. Therefore, similarities and particularities of the metrics cannot be discussed. Contrary to the modularity metrics hardly any contribution show implication of the metric values ([MARTI 2007] and [SOSA ET AL. 2007B] are two notable exceptions).

The contributions to complexity measurement show some undesirable tendencies in structural analysis research: definition of new analysis tools and introduction of new names. For example [AMERI ET AL. 2009] define two new metrics. Although they discuss several other metrics they do not compare them to the new metrics to determine, which metric is the best.

3.4 Product decomposition and modularisation

Product decomposition and modularisation both aim at improving the development and production of a product by defining subsets of a product, which can be handled independently. For example [LAPP & GOLAY 1997] estimate cost savings of up to 15% through modular power plant designs. [ULRICH 1995] considers modular structures as the basis for standardisation and all its benefits. Product decomposition and modularisation are partial solution of the research field product family design. [GERSHENSON ET AL. 2007] propose a research roadmap with modularisation as a key building block in single product design.

Product decomposition is the activity of breaking a product down into modules and components. Product modularisation is the activity of grouping components into modules. Therefore, decomposition is the top-down approach and modularisation is the bottom-up approach for the same task. [SHARMAN ET AL. 2002B] define the basics terms concerning modularisation. This section loosely follows their terminology. Both decomposition and modularisation usually rely on visualisation rather than metrics. The standard structural approaches are matrix reordering and clustering. This section focuses on clustering of a component network. Other methods are introduced but not discussed in detail. [HOLMQVIST & PERSSON 2003] compare six modularisation approaches including matrix-based methods. They point out that no approach handles complex products appropriately.

Most contributions focus on component clustering. Yet, some also address the link between component and team structure e.g. [PIMMLER & EPPINGER 1994] or focus entirely on team structures e.g. [GUTIERREZ 1998]. [KUSIAK 2002] proposes the simultaneous clustering of products and processes to realise all benefits of modular designs. [SHARMAN ET AL. 2002A] introduce multi-domain clustering, which addresses the product, organisation and process domains simultaneously. In particular they show that clustering, which focusses on one domain results in poor clusters in the other domains. [MACCORMACK & RUSNAK 2004] apply clustering to software structures. [DO & CARIGNAN 2005] extend clustering to requirements handling.

[ZAKARIAN 2001] clusters interacting functions to form functional modules. [BAUER ET AL. 2011] point out the importance of matching model definition to the modularisation goals and give an overview of potential domains and goals.

[HARTIGAN 1975] gives an excellent introduction to all variants of clustering. However, this section focuses on clustering of structures rather than collections of data sets. The aim of most structural clustering approaches is to find a visualisation of the structure, which highlights potential modules. In most cases the structure is represented as a matrix with the modules along the diagonal. [SHARMAN ET AL. 2002B] show the limitations of the matrix visualisation and suggest molecular and Venn diagrams to overcome the limitations. Clustering approaches differ in four aspects: target structure (i.e. what the result should highlight), objective function (i.e. how good the result is), clustering algorithm (i.e. how the result is derived) and input data (i.e. what data is used to achieve the result). The remaining section discusses each aspect in detail.

3.4.1 Target structures for clustering

Several approaches for target structures exist. [MCCORMICK ET AL. 1972] propose a block diagonal target structure where all relations form a string of rectangular blocks along the diagonal. Later contributions follow McCormick et al. e.g. [PIMMLER & EPPINGER 1994] and [ZAKARIAN 2001]. [ULRICH 1995] distinguishes three modular structures: slot, bus and sectional. Slot modules have individual interfaces to a central platform and cannot be interchanged. Bus modules have similar interfaces to a central platform and can be interchanged. Sectional structures have no central platform yet similar interfaces. [HELO & HILMOLA 2003] transfer these definitions to software structure. [GUTIERREZ 1998] allows for overlapping clusters.

[SHARMAN ET AL. 2002B] point out that buses (i.e. elements that share relations with many or most other elements) exist in most products and that clustering approaches must deal with them. [GREINER ET AL. 2007] discuss the special role of leaf nodes (i.e. elements with one neighbour) and transit nodes (i.e. elements with two neighbours) for carry-over part considerations. According to them these nodes exist in most products. They recommend removing these elements before clustering.

[LI & CHEN 2008] present various ideal target structures for both inter- and cross-domain matrices, which differ mainly in number and position of the bus structures. Though there are several contributions on target structures no publication discussed so far what the ideal structure depends on. It may depend on the type of product, the variant strategy or even the organisational structure. Moreover, no approach tests the structures a priori, if they contain modules or not even though several metrics allows for assessing the degree of modularity in a structure (see subsection 3.3.2). According to [SHARMAN ET AL. 2002B] and [GREINER ET AL. 2007] two classes of components deserve special treatment: the highly linked (i.e. the buses) and the hardly linked (i.e. the leaf transit elements). Both can be removed before applying the clustering algorithm and added again after that at start and end of the matrix. Whether clusters should be disjoint, overlapping or even nested depends on the specific purpose.

3.4.2 Objective functions for clustering

A wide variety of objective functions have been proposed. The most general formulation is: clusters should encapsulate many internal relations and have little or no relations to other clusters. This partially subjective formulation is commonly used in manual clustering. Software-based approaches require formalised objective functions.

[MCCORMICK ET AL. 1972] use bond energy i.e. the number of 2x2 sub matrices where all four relations exist. Maximizing this number results in block structures. [PIMMLER & EPPINGER 1994] propose minimizing the distance of relations from the diagonal. However, this objective function might result in band structures rather than block structures. [GUTIERREZ 1998] uses a cost-based function, which models the coordination cost among design teams. The cost increases if the teams share many clusters, share larger clusters or do not share a cluster at all. [SHARMAN ET AL 2002A] propose a multi-domain objective function, which is a weighted sum of the clustering value in each domain. However, they omit how the values are to be determined. [YU ET AL. 2003] and [YU ET AL. 2007] use the minimal description length (MDL) as objective function. MDL measures the deviation of the clustered structure from an ideal modular structure with the same number of clusters and cluster sizes.

[SANGAL ET AL. 2005] propose a rule-based objective function for software. The rules define the layers of the software structure and penalise violations. The layers are manual input to the systems. [SANGAL & WALDMAN 2005] extend the rules by enforcement protocols in software development tools. [ZACHARIAS & YASSINE 2008] propose the market coverage of variants as objective function for product family design. [BLEES & KRAUSE 2008] point out that the each department and life-cycle stage has its own ideal module structure. They recommend creating these modules for each of them, to analyse the difference and to finally create a compromise module structure. So far no objective function has gained wide-spread acceptance. Though some authors (e.g. [YASSINE 2011]) discuss several functions no quantitative comparison is available. Moreover, the functions are often subjective as in most cases of manual clustering.

3.4.3 Clustering algorithms

There are two main classes of clustering algorithms: manual clustering and software-based clustering, which includes deterministic algorithms, heuristics and stochastic (in particular genetic) algorithms. The latter three are supported by software. The manual approach is strongly limited by the size of the structure.

[BRADY 2002] manually clusters structures of up to 40 elements. Whereas [RUSHTON ET AL. 2002] claim that their software can deal with up to 10.000 elements. [ZAKARIAN 2001] proposes a semi-manual algorithm, which repeatedly selects one element and group its neighbours into a cluster. Both number and size of the clusters depend on the judgment of the user. [SHARMAN ET AL. 2002B] use manual clustering, as automated approaches were unable “to obtain useful results”. They claim that any automated results require manual intervention. But they also point that manual clustering varies due to the (not least esthetical) preferences of the user.

[SANGAL ET AL. 2005] propose a mixed clustering method for software structures, which contains manual optimisation. [HELMER ET AL. 2010] discuss the manual post-processing of

clustering results in details. In particular they present several ways to deal with bus type clusters and recommend testing the proposed modules against technical feasibility.

[MCCORMICK ET AL. 1972] propose a deterministic algorithm based on systematic positioning and testing all elements of the structure. Though it is suboptimal the results are derived fast and rather close to the optimum. They also discuss an algorithm, which tests all matrix orders but dismiss it as too time consuming for large structures. [HARTIGAN 1975] gives an overview of various algorithms. [VON LUXBURG 2007] gives an overview on spectral clustering, which uses eigenvalues and eigenvectors of adjacency matrices for clustering. The approach is a pre-processing step before applying standard clustering algorithms. [PIMMLER & EPPINGER 1994] use a heuristic based on element swapping but discuss it not in detail. [GUTIERREZ 1998] uses a stochastic algorithm, which picks the elements randomly and optimizes their position. [YU ET AL. 2003] and [YU ET AL. 2007] propose a clustering algorithm based on genetic algorithms. [HELMER ET AL. 2010] modify the algorithm by Yu et al. with random keys. No approach uses modularity metrics to position the elements or to create an initial ordering. Though a wide variety of automated clustering approaches exists many authors recommend to use manual clustering or to optimise the clustering results manually. One reason may be that no objective function captures all aspects of the modularisation problem.

3.4.4 Network models for component clustering

Most contributions use a structural model with one type of relation and one type of element and most clustering algorithms can only deal with such models. When using models with multiple types of relations and multiple types of elements, the models have to be adapted to fit the clustering algorithms. [PIMMLER & EPPINGER 1994] use a weighted component network with four types of relations. They suggest to consider all relations at the same time and to put a stronger focus on spatial relations. [ARTS ET AL. 2008] use the same basic types of relations as Pimmler and Eppinger but extend the model by scenarios of future product adaptations. They claim to create modules, which are more robust against future changes. Based on the work by Pimmler and Eppinger [HELMER ET AL. 2010] propose a scheme for merging the four types of relations into one general type of relation. [GREINER ET AL. 2007] use three views for clustering: geometry, functions and product characteristics. They recommend to start with clustering the functions view as it has the highest impact on verification, testing and quality assurance. [LI & CHEN 2008] propose a framework for handling inter-domain and cross-domain relations simultaneously. There is no consensus how to deal with multiple views on a product for clustering.

This discussion of modularisation omits the definition of platforms as part of product family design. Though there are numerous contributions, which use structural models (e.g. [KALLIGEROS ET AL. 2006], [STEVA ET AL. 2006], [CHENG 2007] or [ZACHARIAS & YASSINE 2008]) no contribution uses structural analysis for the actual definition of platforms.

The contributions to product decomposition and modularisation show some undesirable tendencies in structural analysis research. First, there is hardly any comparison among the approaches. Though each contribution convincingly argues the need for additional clustering approaches hardly anyone tests their results against the state of the art. This may be due to the

lack of consensus on target structure, objective functions and data input. Another reason is the slight predominance of manual and therefore subjective approaches, which deny objective comparison. Second, the approaches are not tested on multiple models. Most contributions provide only single case studies. Thus, they do not prove the applicability to a wide range of problems. Third, the clustering approaches do not use related results from other areas of structural analysis. In particular the results on modularity assessment are not used.

3.5 Description of product and component changes

Change propagation is the phenomenon that changes cause further changes [ECKERT ET AL. 2004], [JARRATT ET AL. 2011]. It is closely linked to the product architecture in particular the component design and the interfaces. As [ULRICH 1995] put it: “product architectures determine how the product can be changed.” The relations among the product components determine potential knock-on changes and their propagation. Therefore, structural models are commonly used to model and analyse change propagation (“DSMs indicate how changes may propagate” [CLARKSON ET AL. 2004]). The research falls into three main categories: empirical studies to assess the characteristics, change prediction and design strategies to encapsulate or limit change propagation. The latter is closely linked to the creation of flexibility and its assessment (“embedding flexibility suppresses change propagation” [DE WECK & SUH 2006] – see also subsection 3.2.3).

3.5.1 Structural models describing change propagation

Most research on change propagation relies on component networks. [CLARKSON ET AL. 2004] give a basic model of change propagation due to physical (i.e. hardware) relations. They also show that purely structural models cannot capture all characteristics of change propagation e.g. the likelihood and the impact of a change. [MACCORMACK & RUSNAK 2004] model change propagation in software products. [GIFFIN 2007] creates a network model of change request, which is derived from a change database and then mapped into a network of software components and a network of engineers. [ARIYO ET AL. 2007] extend the model by [CLARKSON ET AL. 2004] to allow for multiple levels of granularity. [KOH ET AL. 2008] transfer change propagation in a component network to a network of product features, which allows for modelling the improvement of technical attributes. [KOH ET AL. 2009] generalise the model by [CLARKSON ET AL. 2004] to allow for multiple domains and domain-spanning change propagation. [KOH ET AL. 2012] model the impact of changes on customer requirements and [YANG & DUAN 2012] focus purely on parameter networks. [CHUA & ASLAM HOSSAIN 2012] describe change propagation in an activity network and link it to schedule model. [SMALING & DE WECK 2007] propose the Δ DSM to model the change of the product structure due to technology infusion. They also classify the changes, which a product structure can face. Starting from component network of hardware products the models for describing change propagation now allow for multiple-domains and multiple levels of abstraction (see also [HAMRAZ ET AL. 2012]). As the change propagation is mostly researched via simulation most models incorporate additional data like change likelihood, change impact, duration and schedule data.

3.5.2 Empirical research on change propagation

The empirical studies on change propagation aim at describing the phenomenon, its extent and its guiding rules. [CLARKSON ET AL. 2004] show the need for change prediction in a case study based on interviews. Beside the economic impact of changes they show that change propagation of up to four steps occurs. [GIFFIN 2007] analyses 41,000 change requests for software and derives a change propagation network. The network has two important characteristics: no change path is longer than five (and hardly any is longer than four); and, the network forms several independent parts (one of them rather large). Based on these observations a classification scheme for software modules is proposed: absorber, multiplier or constant. [GIFFIN ET AL. 2009] extend this work using motif analysis and propose a normalised change propagation index. [PASQUAL & DE WECK 2012] continue the analysis of the dataset but focuses on the social impact of change propagations. They show that change propagation often involves system interfaces (i.e. interfaces among different organisational units) but never involves more than three system interfaces. In the dataset 22% of all changes propagate indirectly. [SHANKAR ET AL. 2012] present a case study from the automotive industry. Their focus is the change reason. In the study about a third of the changes (32.4%) propagate. The main reasons for change propagation are document error rectification and design error rectification – most of them are triggered internally (77%).

Most of the empirical studies rely on the retrospective analysis of change management databases and rely on the accuracy of the change documents. Though the case studies deal with huge datasets the generalizability of the results is unclear. The findings sum up as follows: most changes (about two thirds) do not propagate; most changes (about three quarters) are triggered internally; the change paths are not longer than four (or five in exceptional cases); change propagation involves mostly one or two (more than three quarters of the cases) and never more than four organisational units.

3.5.3 Simulation approaches for change prediction

Change prediction relies on two main approaches: scenario-based and simulation-based change prediction. The scenario approach anticipates changes and estimates their impact. The contributions differ in terms of change extent ranging from “what-if-scenarios” (e.g. [KELLER ET AL. 2005], [WILDS 2008] and [KOH ET AL. 2012]) to technology infusion (e.g. [SMALING & DE WECK 2007] and [SUH ET AL. 2008]). Some authors also propose planning tools based on change propagation e.g. implementation and release planning of software components [NORD ET AL. 2011]. [HERFELD ET AL. 2007] propose to use search strategies to find the root causes of changes. All simulation-based approaches extend the structural models by additional data or link them to other models. [CLARKSON ET AL. 2004] propose the change prediction method. It uses change likelihood, change impact and change duration as additional input to create simulation models. Several extensions of the model (e.g. [ARIYO ET AL. 2007], [KOH ET AL. 2009] and [KOH ET AL. 2012]) add new domains to the model but do not vary the simulation model itself. Both, [KELLER ET AL. 2009] and [KOH & CLARKSON 2009], present case studies on the change prediction method and its feasibility. [KELLER ET AL. 2006B] show that the change prediction method is equivalent to disease spread models. Moreover, they show that the simulation algorithm does not impact the results qualitatively i.e. the implications of the results

do not differ. [CHUA & ASLAM HOSSAIN 2012] simulate change processes and link the propagation model to a schedule model.

Though various approach exist the have hardly been tested in practice i.e. the predictions have not been compared to later actual changes. Moreover, no prediction method has been tested against empirical studies i.e. if the scenarios or simulations can reproduce the observations. There are two reasons for this. First, the scenarios and simulations cover more potential changes than can occur in reality. Second, one of the main reasons for change prediction is avoiding change by better designs or preventive actions.

3.5.4 Design strategies for limiting change propagation

The research on design strategies aims at encapsulating or limiting change propagation. It focuses on the early design stages in particular concept or architecture design. [ULRICH 1995] states: “product architecture determines how the product can be changed.” Flexible design solutions allow for “suppressing change propagation” [DE WECK & SUH 2006]. However, flexible designs are often costlier. This conflict is reflected in the main applications of change prediction: familiarisation, risk identification, identification of propagation absorbers/multipliers and testing of solutions [KELLER ET AL. 2009]. The results of change prediction provide guidance for the preliminary product design [KOH & CLARKSON 2009]. However, the solutions must also be tested against cost and benefit [NORD ET AL. 2011] as well as schedule considerations [CHUA & ASLAM HOSSAIN 2012].

The research on change propagation is split into two areas. The first area deals with the observation of change propagation, its frequency, its impact and its rules. Most of the results rely on huge datasets derived from change management tools. However, the bigger area deals with change prediction and avoidance. Contrary to the first area it mostly uses theoretical and simulation approaches. The reason is that many more change are possible than really occur. The aim of change prediction and avoidance is to create product designs, which are robust to changes. The two areas are not linked in particular it has not been tested whether the prediction methods reproduce the observations.

3.6 Conclusion

The purpose of this chapter is to determine, which structural analyses are applied to product and component models and how they are validated and evaluated in research. Thereby, the current challenges and weaknesses in research are revealed and can be approached in this thesis.

Section 3.1 of this chapter gives a short introduction to the theoretical background of structural analysis. It showed that there is an almost infinite amount of analyses available due to recent advances in graph theory such as motif analysis. Next, the structural analyses are classified and the major types are defined. Finally the scope of this thesis in term of structural analysis is given: it focuses on the metrics in structural analysis and omits further research on graph theoretic algorithms. Moreover, the thesis does not deal with visualisation.

Section 3.2 introduces the structural analysis of life-cycle properties. The aim is to identify system elements, which are important to the system behaviour during its life-cycle. Most life-

cycle properties are associated with modifications of the product during its development, production and usage. Metrics have been proposed for robustness, changeability, flexibility and adaptability of product components. Many approaches use structural models for dealing with or measuring life-cycle properties. Most of them do not use purely structural models but require additional data such as cost, impact or likelihood estimations. Though there is some evidence that the product structure impacts the life-cycle properties there are no empirical long-term studies on how strong or valuable that impact is. Moreover, the individual properties are closely linked both among themselves and to concepts like change propagation or modularisation.

Section 3.3 introduces the structural analysis of product-inherent properties. The aim is to choose appropriate product concepts and to find ways to improve them. Most product-inherent properties cannot be assigned to or broken down to individual components like weight but result from the interaction of all components like power consumption. Therefore, most approaches evaluate entire product component networks. Research in structural analysis on these emergent properties focused on quality, modularity and complexity. No approach measures the properties directly. They rather assess the consequences of high or low quality. Or, in case of modularity: they test what distinguishes products, which are accepted to be modular from products, which are accepted to be integral. Similar to the modularity metrics no approach measures complexity directly as no quantifiable definition of complexity has been proposed so far. Most empirical results stem from the analysis of software products. As discussed in section 2.2 software structures can be easily derived from the source code. If the results for software can be transferred to hardware is still an open research question (see sections 2.1 and 2.2 for a discussion). The contributions also show that researchers tend to define new analysis tools, develop new computation methods and introduce new names. This leads to little comparison of the approaches.

The sections 3.4 and 3.5 describe modularisation and change propagation. For both of them manual and automatic approaches exist. And, both of them support the creation of product concepts, which robust and flexible towards changes and allow for fast, cost-efficient development. Moreover, both fields of structural research show similar weaknesses. First, there is hardly any comparison among the approaches. Each contribution claims the need for research hardly anyone tests their results against the state of the art. Second, the approaches are not tested on multiple models. Most contributions provide only single case studies. Thus, they do not prove the applicability to a wide range of problems. Third, there is a lack of empirical testing the predictions and results. The results are mostly confirmed by interviews and no long term studies exist. The lack of comparative and empirical studies results from the experimental challenges, in particular the high effort for data collection and model creation.

Though many more analyses for hardware models have been introduced, the analyses for software are usually more reliable due to the applied research methods. These observations result in several open research tasks (the first two are addressed in this thesis):

- narrow down the almost infinite set of analyses to a pragmatic and meaningful set
- consolidate research on the assessment of life-cycle and product-inherent properties
- empirical research on modularisation and change propagation to verify the predictions

This thesis provides criteria to narrow down the set of analysis based on theoretical and empirical work. It focuses on structural metrics and the assessment of life-cycle and product-inherent properties.

4. Improving structural analyses

This chapter describes the approach to improve structural analyses namely the reduction structural metrics to a minimal set. Previous chapters introduced the aims of this thesis (chapter 1) and gave an introduction to structural modelling (chapter 2) and metric-based structural analysis (chapter 3). The next chapter presents the research methodology to implement and test the approach.

The next section (4.1) describes the validity criteria for structural analyses. The main criteria are computability, disparity and significance. Computability is a rather qualitative criterion, which is usually fulfilled and is often not explicitly tested. Disparity means that metrics must have a suitable value ranges and differ significantly from each other. It is hardly tested at all. Testing this criterion for product component structures is the main contribution of this thesis. Significance is the most important criterion for the practical applicability of structural analysis. The requirement is fulfilled if a metric allows for describing or estimating a relevant system property. Section 4.2 focuses on the key task and challenge of research on structural analysis: showing the significance of an analysis. Most research so far relied on qualitative methods like analogy and comparison. The most reliable results originate from empirical research and to a smaller degree from simulation. The section shortly describes each method, gives an example from literature and highlights the key challenges.

Section 4.3 proposes a scheme for describing and documenting structural analyses called “structural analysis scenarios”. The scenarios comprise all information to perform sensible structural analyses. They include definitions, guidelines and hints for modelling, analysing and interpreting a structural model. The results of structural analysis depend on the aim, the type of model and the metric, which are the constituting elements of structural analysis scenarios.

The final section of this chapter (4.4) introduces the research questions and hypotheses of this thesis. The main hypothesis: structural analysis metrics are highly redundant and can therefore be reduced to a smaller set with the same analytical power. The remaining thesis describes how this hypothesis is tested and validated.

4.1 Validity criteria of structural analyses

The section describes the validity criteria for structural analyses: computability, disparity and significance – an early version of the criteria is introduced in [BIEDERMANN & LINDEMANN 2011B]. The criteria were defined based on the literature, discussions with experienced users of structural analysis and the working experience of the author. Computability (see subsection 4.1.1) means that an analysis can be performed on a given model – it comprises the definition, the algorithm, the implementation and the running of the analysis. Computability can be tested with a checklist and is usually fulfilled. Disparity (see subsection 4.1.2) means that metrics must have a suitable value ranges and differ significantly from each other. Disparity can be tested with a meta-analysis of a class of models. The criterion is tested in this thesis for structural metrics for product component networks. The methodology is presented in chapter 5. Significance (see subsection 4.1.3) means that the analysis allows for solving the initial

problem. Significance must be tested for each combination of problem, type of model and analysis – this test is main task and challenge of research. Various methods for testing the significance of a structural analysis have been presented in the literature. Section 4.2 introduces the most important methods and describes them in detail. Figure 4-1 gives an overview of the validity criteria and their main testing methods.

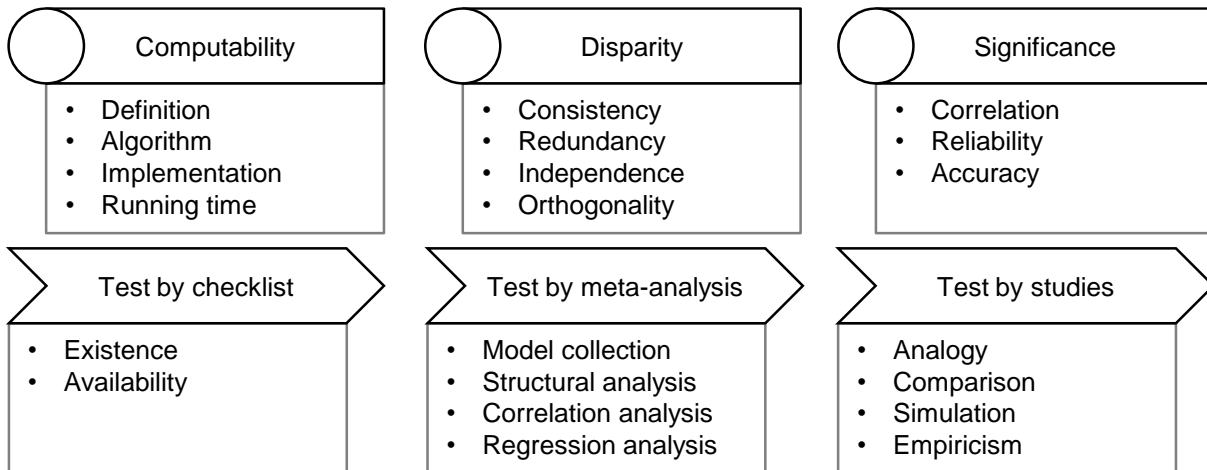


Figure 4-1 Criteria for the applicability of structural analysis and methods to test the criteria

The fulfilment of computability and disparity are necessary yet, not sufficient criteria for structural analysis as is explained in the next subsections. Though significance is sufficient it may lead to undesirable consequences for structural analysis. A new metric may be significant for solving a design problem. However, it may also be redundant to another already available metric. Therefore, testing the disparity criterion is also crucial for sensible and efficient structural analyses.

4.1.1 Computability

The basic requirement is computability of the metrics based on the structural model. It comprises two sub-requirements. First, the metric must be defined for the type of structure and an algorithm to compute them must be known and implemented. Metrics can be defined for undirected structures only (e.g. blocks) or for directed structures only (e.g. active degree). Directed structures can be transformed into undirected ones to allow for applying all structural analyses. Transforming undirected to directed structures is not feasible as the results do not reflect the directedness. Second, the metric must be computable in a given time. The computation time depends on the complexity of the structure, the complexity of the algorithm, the implementation of the algorithm and the computer hardware. The available computation time depends on the project and the analysing engineer.

There are many tools available for structural analysis (see e.g. [WYNN ET AL. 2009] or [LAU 2006]). The issue of implementation therefore hardly arises. The computation time is not limiting the application of structural analyses in engineering design due to advances in computer hardware and algorithmic graph theory. In the publications describing applications in

engineering design the computability of structural analyses is not addressed unless many models are involved (see e.g. [BROWNING & YASSINE 2010A]). In network theory much larger structures are analysed and computation time is still an issue [CAMI & DEO 2008]. There is no specific testing method for this requirement. It is usually not explicitly tested.

4.1.2 Disparity

The disparity criterion refers to the values, which structural metrics may have. In contrast to the computability it depends on the application and the type of structure. The fulfilment of the requirement depends on the system, on the type of structure and the structural model. There are two main types of disparity. First, the values of a metric must vary in models of real products. Depending on scope of the metric this applies to components (for local metrics) within a product or networks of several products (for global metrics). Second, the metric values must not correlate to the values of other metrics. More formally expressed: the metrics must be orthogonal both in theory and in practice.

Variation of values in a component network: The values of the metrics must vary among the elements of the network. The rarest and the most extreme forms of the analyses (e.g. the highest degree) characterise elements, which have outstanding importance for the system (see e.g. [KREIMEYER & LINDEMANN 2011] for a discussion of this concept). The requirement can be tested within one system model. To prove the general applicability several models have to be tested. In literature this requirement is mostly not explicitly addressed. Some metrics may not vary or show invariant upper or lower bounds (e.g. the degree is almost never lower than one in component networks). These metrics provide characteristics of the product and allow for detecting modelling errors (see e.g. [SCHMITZ ET AL. 2011]). [BRAHA & BAR-YAM 2007] provide an example for activity networks. They show that there is an upper bound for the passive degree and assume that the bound results from cognitive limitations. However, finding the invariants requires the analysis and comparison of many networks.

Variation of values among product networks: This requirement applies to structural metrics for comparing product concepts e.g. by estimating product-inherent properties. The values of the metrics must vary across systems of the same type. The requirement can be tested by comparing a few system models. In the literature this requirement is hardly addressed. It is expected to be fulfilled. If the values of a metric do not vary (significantly) in several models this value may be an invariant of the models and be typical for the type of systems. For example [WHITNEY 2011] observe that components networks of mature products have an average degree of about 6.3. They conclude that this is typical for mature systems and propose the average degree as a measure for the maturity of a product design. Invariant metrics in general could be used for model testing. However, finding the invariants requires the analysis and comparison of many component networks.

Orthogonal metrics: The metrics must not correlate to each other. Otherwise the metrics are redundant and provide no new insights. To show orthogonality it must be proved that the metrics cannot be derived from each other. This requires a wide range of models. The models are structurally analysed. The results are then statistically analysed e.g. by correlation analysis

or regression analysis. This thesis focuses on deriving a set of orthogonal structural metrics. The research methodology follows the idea above and is described in detail in chapter 5.

4.1.3 Significance

Significance is the most important applicability criterion. The fulfilment of the two other groups of criteria is necessary but not sufficient for a metric to be applicable. The requirement is fulfilled if the metric allows for describing or estimating a relevant system property. If the purpose is reduction of development time the metric must e.g. correlate with the process duration. If the purpose is increasing product quality the criterion must e.g. correlate with error or failure frequencies. A wide range of testing methods have applied and described in literature. The next section introduces four methods for testing and proving the fulfilment ranging from purely theoretical to empiric methods.

4.2 Methods for showing the significance of structure analyses

Analogy, comparison, simulation and empiricism are the main methods for showing the significance described in literature. All of them have been used in literature – in most cases analogy. They differ in three correlated dimensions: qualitative-quantitative, lenient-rigorous and theoretical-empirical. The choice of method mainly depends on the availability of data. The more data available is the more quantitative, rigorous and empirical the method of choice is.

Properties of the methods			Method	Idea	Main challenge
<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">qualitative</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">lenient</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">theoretical</div> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">quantitative</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">rigorous</div> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">empirical</div> </div>	Analogy	Describe a similarity between the structure and a known phenomenon.	Finding the analogy.		
	Comparison	Define ideal reference structures and measure the difference between a real and the reference structures.	Defining the reference structures.		
	Simulation	Create a structural and a simulation model of a system and compare the structure with the simulation results	Proving the validity of the simulation results.		
	Empiricism	Compare the results of structural analysis with independently derived observations.	Creating the data and proving independent acquisition.		

Figure 4-2 Methods for showing the significance of structural analysis

Analogy (0) connects an analysis and a phenomenon via theoretical discourse and requires no further data. Comparison (4.2.2) connects an analysis and a phenomenon via reference structures, which represent extreme forms of the phenomenon. Simulation (4.2.3) connects an

analysis and a phenomenon via simulation of the phenomenon based on a given structure. Empiricism (4.2.4) connects an analysis and a phenomenon via simultaneous observation of both. Each subsection presents a short definition of each method, an example from literature and the main challenges when applying the method. Figure 4-2 gives overview of the methods including their main characteristics, their main ideas and the main challenge.

4.2.1 Analogy

This approach builds an analogy between a metric (e.g. number of cycles per edge) and a known phenomenon (e.g. iterations in design process). The implications, properties and effects (e.g. uncertainty of the process duration) of the phenomenon are transferred to the metric. The significance and relevance of the metric is correlated with the phenomenon's properties. This method only requires intellectual work and results in a line of arguments, which connect a phenomenon with the metric.

[KUSIAK & WANG 1995] use this approach to develop a structure-based sequencing method. They describe an analogy between iterations and cycles in process activity networks. Iterations are repetitions of activities and tasks. Cycles are close sequences of information flows among activities. Iterations tend to increase the process duration and the planning uncertainty. Therefore, a high number of cycles indicates process uncertainty due to iterations. Efficient dealing with cycles allows for better handling of iterations and removal of cycles lowers the risk of iterations. However, they do not provide empirical evidence that many cycles indicate a lot of iterations.

The analogy approach does not allow for quantified structural analyses as only tendencies but not quantified parameters are inherited. Even comparisons among structures of similar type are hardly possible. For example, when comparing two process models – one containing twenty cycles; the other containing thirty cycles – it is impossible to predict duration uncertainty or even if and how much more uncertain the duration of the second process is.

This approach is particularly suited if there is hardly any data available but some theoretical work (including the definition of the main terms) exists.

4.2.2 Comparison

In this approach exemplary structures are created, which possess extreme structural properties (e.g. purely integral and purely modular). They are expected to represent ideal systems with pure characteristics (i. e. without trade-offs as they occur in real engineering systems). The structures are compared to real systems. The differences can be quantified to measure the real system's properties in relation to the ideal systems. This approach requires the definition of the ideal structure. They are the result of theoretical work quite similar to the process in analogy. Alternatively, the structures of two real systems with different characteristics (e.g. one rather integral and one rather modular) are compared and the structural differences are highlighted. However, this second approach requires that the characteristics of the systems are generally accepted.

[HÖLTTÄ-OTTO & DE WECK 2007] use this approach to measure the degree of modularity of engineering systems and products. They define three exemplary (or canonical in their terms) structures: integral, bus-modular and modular. They compare these with real system structures. They also compare pairs of systems with the same functionality but different technological constraints e.g. cell phones and desk phones. Highly-constraint systems tend to be more integrally modularised. Hölttä-Otto and de Weck also include random structures to show that real engineering systems have significantly different structural characteristics. Comparing real structures to randomly created ones is a common research approach in network theory [CAMI & DEO 2008]. [HÖLTTÄ-OTTO & DE WECK 2007] use this approach as modularity still lacks a formal (i.e. measurable) definition. Therefore, they have to use ideal structure and “twin” structures to prove the significance of their metrics.

The approach allows for semi-quantified result. They are limited to measuring the differences to the exemplary structures. They cannot measure system properties directly as the properties of the exemplary structures are not quantified. In the example the resulting metrics allow for determining the more modular system but not for measuring an absolute degree of modularity. The main challenges are to find appropriate reference structures, and to reliably determine the real structures.

This approach is particularly suited for research of emergent features such as complexity and modularity. Usually a formal definition is lacking, yet most engineers agree on the characteristics of certain examples.

4.2.3 Simulation

In this approach simulation models (e.g. progress of design projects) are derived from structural models (e.g. activity networks). The simulation results (e.g. project duration) are compared to structural metrics (e.g. relational density). The significance and relevance of metrics are shown by correlating them with significant and relevant simulation results.

[BROWNING & YASSINE 2010B] use this approach to evaluate priority rules for resource allocation in multi-project environments. They show that relational density of activity networks is one of three criteria to choose appropriate priority rules. They achieve this result by synthesizing (see [BROWNING & YASSINE 2010A]) and simulating 12,320 project set-ups. The variations and means of the simulation results were statistically analysed. The analyses showed a significant correlation between relational density and the appropriate choice of priority rules (i.e. the rules with lowest average project duration).

The simulation approach allows for semi-quantified structural analyses. Comparisons between systems are possible but absolute predictions are generally not. The main challenge is to create simulation models, which cover the complete parameter space. Both, the space of the potential structures and the space of the simulation models have to be explored. Therefore, the approach is rather time-consuming as a lot of simulations have to be run. Moreover, the simulation model must be appropriate and a lot of system structures must be available. In the example the activity networks were synthesised and automatically parameterised to create the simulation models. Therefore, the authors had to show the synthetic structures mimic real activity networks. They use an extensive literature review to collect the typical characteristics of activity networks and

define appropriate ranges for the project parameters. For most types of systems synthesis methods are hardly available. However, there are many contributions for components models in the field of computational design synthesis (see [HELMs & SHEA 2011] for a structure-based approach). As a final remark: the simulation approach uses the structural model as input. Therefore, the results might be tautological i.e. a direct result of the research process rather than a new insight.

This approach is particularly suited if a well-defined theory exists but observations are costly, time-consuming and/or unreliable.

4.2.4 Empiricism

In this approach the statistical relation between structural metrics (e.g. coupling among software components) and system properties (e.g. number of design errors) is determined. If the results are statistically significant the structural analyses are significant as well. The relevance of the analyses depends on the relevance of the system properties.

[SOSA ET AL. 2008] use this approach to show the connection between coupling in the component structure of software systems and the quality of the software system. They analyse the structures of 20 software systems (in 108 versions in total). For each version they compare the number of bugs and the number of resolved bugs with the coupling (e.g. in form of cycles) within the structure of the previous version. They show that high actual coupling (originating from the architecture) increases the number of bugs and that high intrinsic coupling (originating from the organisation of the engineering project) decreases the capability to fix bugs.

The statistical analysis approach allows for quantified results. Even absolute predictions are available provided the research relies on enough data. The four main challenges are: to model enough structures to be statistically significant, to reliably create the structural models, to determine the system properties independently of the structure and to avoid hidden parameter biases. In the example 108 models were created by automatically parsing the source code of each software release. Thereby the modelling effort was rather low and very reliable due to the parsing algorithm (this is typical for creating software models; see section 2.2). The analysis results were compared to the bug reports. These reports are created and documented separately. Therefore, structure and phenomenon were determined separately. In the original contribution hidden parameter biases were not discussed. In [SOSA ET AL. 2011], which is a follow-up contribution, new findings were tested against seventeen control variables to avoid hidden-parameter biases.

This approach is particularly suited if structural models can be easily created and the phenomena of interest can be reliably and efficiently observed.

4.3 Structural analysis scenarios

Practitioners require guidelines for applying of structural analyses. They need to know, which model to create, how to analyse it and how to interpret the results. Therefore, researchers on structural analysis should provide this information if not actual guidelines. This thesis proposes a schema for documenting the guidelines: structural analysis scenarios (see also [BIEDERMANN

& LINDEMANN 2011A]). They document the way of performing an analysis and interpreting the results. The scenarios comprise the initial problem (i.e. the purpose of the analysis), the type of model and the analysis itself. This section presents an approach to make structural analyses ready for industrial application. The previous sections describe the requirements and challenges when researching structural analyses. Though this thesis does not provide a conclusive set of scenarios it consolidates the research on structural metrics for component networks, which is a necessary step when preparing structural analysis scenarios.

The problem (or aim or question) defines the objectives of the analysis including key indicators and the required accuracy of the results. The problem description defines the purposes of the analysis e.g. defining an appropriate module structure or predicting quality problems of singular components. The problem description gives an explanation of the key terms e.g. that appropriate modules should be separately testable. This makes sure that the practitioner chooses the appropriate scenario. Additionally (and for researchers more importantly) the achievable accuracy is documented. Most structural analyses only allow for qualitative comparisons e.g., which component is most likely to be changed. More detailed predictions e.g. how much more likely a change is or even an absolute prediction of the likelihood are mostly not possible. By documenting the accuracy exaggerated expectations are avoided.

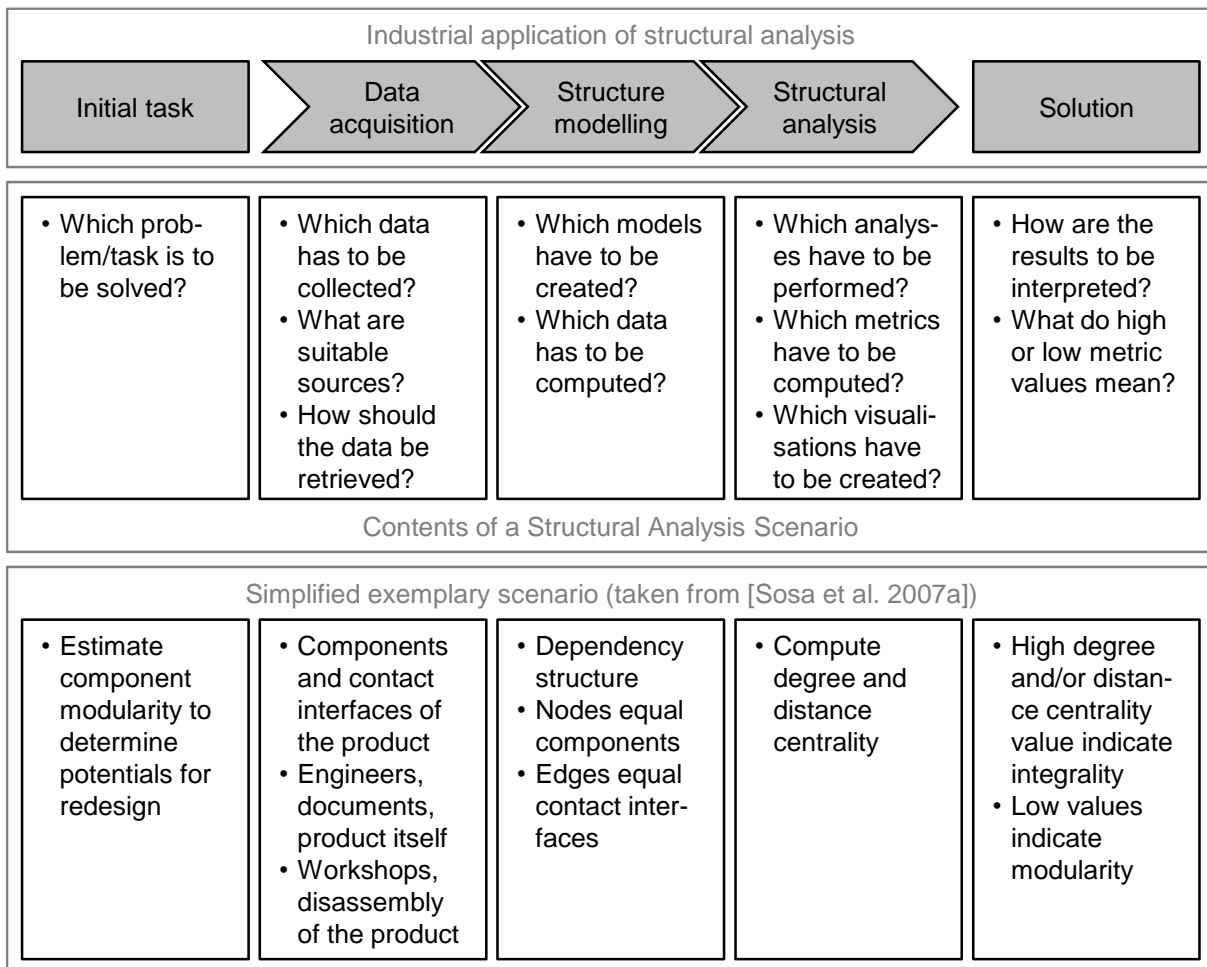


Figure 4-3 Contents of a Structural Analysis Scenario including a simplified example taken from [SOSA ET AL. 2007A]

The description of the type of model includes the type of elements, the type of relations and the appropriate level of abstraction. This defines the model to be created and its fundamental characteristics e.g. directedness of the relations. These characteristics determine the available analyses. Chapter 2 gives an overview on component-related structural models. The scenarios also include recommendations for data sources and acquisition methods. However, each sensible combination of source and method should result in the same model. This may be incorrect in practice. This thesis focuses on the analysis of structural models not their creation.

The description of the analysis includes the definition, the computation, the presentation and the interpretation of the results. The basis of the description is the listing and definition of the analyses e.g. the definition of the degree metric as the number directly connected elements for each element. This allows for understanding and interpreting the results. The computation only lists the available algorithms and tools for the analysis but gives no details like implementation strategies as various toolkits exist (see subsection 4.1.1). The presentation part suggests visualisations of the analyses. In case of metrics mostly diagrams (e.g. portfolios) are recommended. Finally, the interpretation part guides the final step of the analysis: gaining insights from the results. In case of metrics the significance of high and low values as wells as value ranges is provided. Several collections of structural analyses are available and often provide this information e.g. [LINDEMANN ET AL. 2009] and [KREIMEYER & LINDEMANN 2011].

4.4 Minimal analysis sets

The previous sections describe the validity criteria for structural analysis (4.1), the main methods for testing the significance of an analysis (4.2) and a documentation scheme for structural analysis (4.3). This section introduces the main approach for improving structural analysis: fundamental analysis sets. Chapter 3 provides an overview of the current state of the art in structural analysis of component networks. It highlights a lack of comparisons among the analysis approaches in particular of the metric-based approaches – both among the metrics and across multiple systems. Therefore, it is not known if the disparity criterion is fulfilled. These observations lead to three research questions:

- Are the structural metrics mutually independent?
- Are there reduced sets of metrics, which cover the complete analysis space?
- Is there a minimal (i.e. fundamental) set of metrics, which covers the analysis space?

In other terms: is the disparity criterion fulfilled? And if not, is there a set of metrics, which fulfils the criterion. The main contributions of this thesis are the answers to this question for product component structures. Based on the insights from the state of the art chapters 2 and 3 three hypotheses are stated and tested in this thesis:

- Structural metrics strongly interdepend and correlate.
- There are many reduced sets, which cover the analysis space.
- There are several alternative minimal sets.

Figure 4-4 highlights the concept of metric of the minimal set sets. The next chapter introduces the research methodology for testing the hypotheses and defines the main types of dependencies among structural analyses.

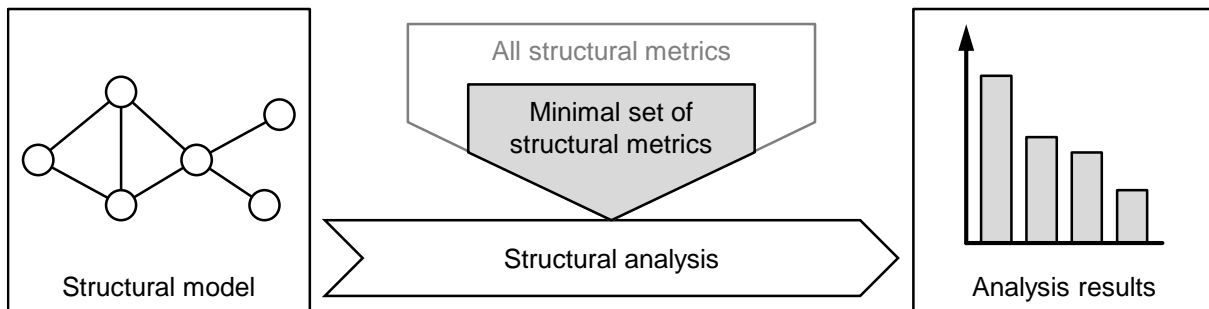


Figure 4-4 Fundamental analysis sets

4.5 Summary

This chapter introduces the approach of this thesis to improve structural analyses. As shown in chapter 1, most industrial applicator request more guidance in modelling and analysing product structures. This thesis aims at improving the analysis part by focusing on analyses, which reliably provide fundamental insights for the applicator.

Criteria for the applicability of structural analyses were introduced in section 4.1: computability, disparity and significance. Computability is only formally relevant as advances in algorithmic graph theory and computer hardware allow for efficiently computing most structural analyses. Disparity has so far hardly been addressed in research (see also chapter 3). Therefore, structural analyses may be redundant or limited to a small number of products. Testing the disparity criterion for structural metrics of hardware components is the focus of the remaining thesis. Significance is the most important criterion and testing it is usually the focus of research contributions to structural analysis.

Section 4.2 describes methods for showing the significance of an analysis. The methods differ in terms of rigor and reliability ranging from qualitative methods like analogy to quantitative methods like empiricism. The more rigorous and reliable a method is the more data it requires. Research contributions to the analysis of the hardware models usually use rather qualitative methods. Whereas contributions to the analysis of software models usually use quantitative methods (see also chapter 3). Though many more analyses for hardware models have been introduced, the analyses for software are usually more reliable due to the applied research methods.

Structural analysis scenarios as guidelines for applying structural analyses were introduced in section 4.3. Each scenario comprises all information a user requires for performing a structural analysis: the initial task; the required data (and how to acquire it); the necessary models, analyses methods and result visualisations; and instructions for interpreting the results and thereby solving the initial task. Ideally, all research contributions to structural analysis should provide this information concisely. However, so far most contributions omitted some details.

Finally, metric of the minimal set sets as the main contribution of this thesis were introduced in section 4.4. The idea is to find a set of few metrics, which provide the same insights as all

available metrics. The research approach of this thesis is to test the disparity criterion for a relevant class of structural models, namely contact relation networks of hardware components. By proving that the metrics are disparate and omitting those, which are not, a minimal set is constructed. The detailed methodology for finding and validating the minimal set is described in the next chapter.

5. Determining the interdependencies of structural criteria

The previous chapter presents the main approach of this thesis for improving structural analyses: defining a minimal set of structural metrics for component networks. This chapter describes the research methodology for finding candidates for the minimal sets and validating them afterwards. Section 1.5 provides an overview of the research methods of the entire thesis (including the state of the art). This chapter provides the details.

First, the main types of relations among structural analyses are presented in section 5.1. There are two kinds of relations among structural analyses: relations resulting from the definition and relations which are characteristic for the type of system. The relations by definition do not reduce the set of metrics as they only prescribe interdependencies among the interpretations. Relations by type of system reduce the analysis set as they result from the system's characteristics. Therefore, the thesis builds on the relation by type of system for finding the minimal sets.

Second, the collection of structural analyses, which are tested for disparity, is shortly introduced in section 5.2. The thesis focuses on structural metrics coming from graph theory rather than motif analysis.

Third, section 5.3 describes the actual research procedure for finding and testing the minimal sets. Basis for identifying metric of the minimal set sets is a correlation analysis of metric values across a representative set of models. The clusters within the correlation relations represent groups of metrics, which can be replaced by one metric from the group as they are broadly equivalent. These representative metrics form one candidate for a minimal set. The proposed minimal set is validated in two studies to show the set has the same analytical power as all tested metrics.

5.1 Taxonomy of relations among structural analyses

There are two broad groups of relations among structural analyses: relations resulting from the definition and relations, which are characteristics for the type of system. Both groups impose limitations on the applicability of the analysis.

The relations by definition are inheritance, composition and derivation. They are a direct consequence of their mathematical definition and have been found by graph theorists. Inheritance means that one analysis is a more specialised type of another analysis. For example, a leaf node is a special type of bi-connected component (see section 11.1 to 11.3 of the appendix for the definitions). Composition means that one analysis is assembled of other analyses. For example, a strong component is composed of cycles. Finally, derivation means that the results of an analysis are derived from the other. For example the distance among two nodes is derived from a shortest path between them. Section 11.4 of the appendix presents a model of the formal relations among structural analyses, which is taken from [BIEDERMANN & LINDEMANN 2011A].

The relations by type of model are coexistence, correlation and exclusion. They were defined in discussions with experienced researchers on structural analyses. Coexistence and exclusion

are closely interrelated. Coexistence means that specific results of two analyses always occur together in a model of a particular type even though there is no formal reason (e.g. a composition relation) for that. Exclusion means that specific results of two analyses never occur together in a model of a particular type even though there is no formal reason for that. Both relations have not been observed so far. Therefore, no example is presented. Correlation means that two analyses (in particular metrics) result in values with a seemingly linear relation. For example, high values of the degree metric correlate with a high number of cycles containing that node (the same applies to the entire value range).

The relations by definition allow for checking consistency and redundancy among the interpretations of the analyses. The idea behind this: the interpretations of two formally interlinked analyses must not contradict each other. For example, a strong component is composed of cycles. Therefore, the interpretations of the two must be consistent in each context and for each type of model. In process networks cycles indicate potential iterations; strong components indicate process phases with encapsulate iterations. The interpretations are consistent and not redundant. Figure 5-1 shows this example. The relations by definition allow for consistency checks but not for a reduction of the set of analyses. Section 11.4 of the appendix describes the consistency checks in more detail (based on [LINDEMANN & BIEDERMANN 2011A]).

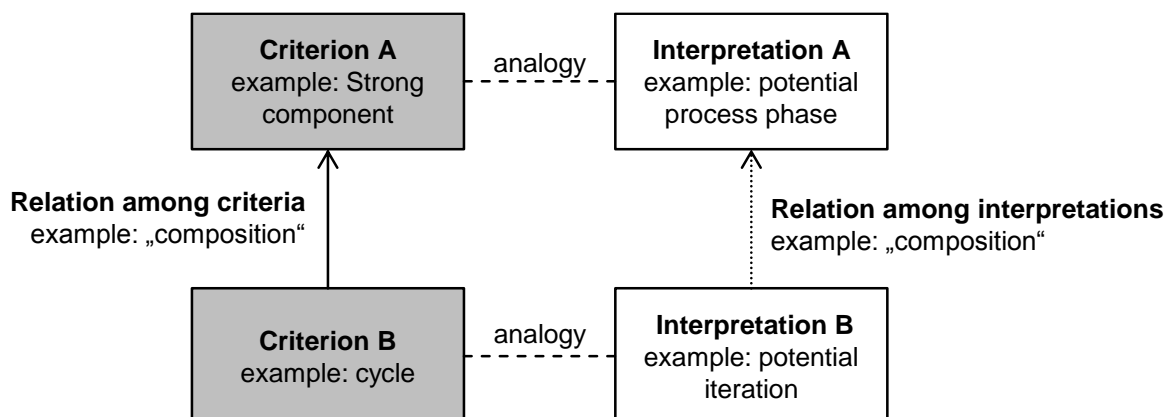


Figure 5-1 Dependencies of the significances due to relations among the structural analyses

Relations by type of model allow for reducing the set of analyses by replacing a group of highly correlated analyses by a single analysis. The idea behind this: highly correlated analyses provide the same basic insights; therefore only one of them is needed. For example, if degree and number of cycles are highly correlated in component networks both or only one of them allow for determining highly constraint components. For the purpose of reducing the analyses set positive and negative correlations are equally suited.

Both groups of relations impose limitations on the applicability of the analysis. However, relations by definition mostly address the interpretations of the analyses. Hence, they allow for checking the consistency and redundancy among the analysis but not for reducing the set of sensible analyses. In contrast, the relations by type of system address the actual results of the

analyses in real systems. Therefore, they (in particular the correlation type) are suited for finding minimal sets.

5.2 Collection of structural analyses

This thesis focuses on structural metrics and omits properties, subsets and visualisations of structures (see subsection 3.1.2. for a classification of structural analysis. Most metrics are taken from the state of the art and graph theory. Metrics from motif analysis are omitted as they are far too numerous and mostly are special types of the graph-theoretical metrics.

Previous works proposed several collections of structural analyses. The collections are not applicable to component networks. [KREIMEYER & LINDEMANN 2011] focus on metrics for process networks. These networks are usually directed, whereas component networks are mostly undirected. Therefore, the metrics are not applicable (see subsection 4.1.1). [LINDEMANN ET AL. 2009] provide an extensive collection of structural analyses. However, they are only described for two general (and rather abstract) classes of structures: essentially directed and undirected ones. Both collections partially contributed the collection of metrics for this thesis.

Table 5-1 Structural metric collections

Global structural metrics	Local structural metrics
Average degree (X)	Average distance to node (X)
Average distance centrality (X)	Blocks per node (L)
Average number of blocks per node (X)	Cliques per node (L)
Average number of cliques per node (X)	Clustering coefficient (L)
Average number of cycles per node (X)	Cycles per node (L)
Average path centrality (X)	Degree (X)
Average path length (X)	Distance centrality (L)
Number of blocks (L)	Maximum distance to node (X)
Number of cliques (L)	Median distance to node (X)
Number of components (L)	Path centrality (L)
Number of cycles (L)	
Number of edges (X)	
Number of nodes (X)	

L: computed by LOOMEOTM 2.5.0

X: computed by Microsoft ExcelTM 2010

The final collection (see Table 5-1 for the complete collection) comprises thirteen metrics for whole networks (so-called global metrics) and ten metrics for elements within networks (so-called local metrics). The collection is selected from previous collections and the literature review. The global metrics fall in two categories: averages of local metrics e.g. the average degree and total numbers of structural subsets e.g. number of cycles. The local metrics are more differentiated: centrality metrics e.g. the degree, distance metrics e.g. maximum distance to node and frequency metrics e.g. cycles per node. The formal definitions for the metrics are in section 11.2 of the appendix for global metrics and in section 11.3 for local metrics.

5.3 Procedure to determine metric of the minimal set sets

The basic idea for identifying metric of the minimal set sets is a correlation analysis of metric values across a representative set of models. The clusters within the correlation matrices represent groups of metrics, which can be replaced by one metrics from the group. These representative metrics form a candidate for a minimal set. The proposed minimal set is validated in case studies in order to show, that the set has the same analytical power as all metrics.

The research methodology of this thesis follows a five-step-procedure, which is shown in Figure 5-2. First, models are collected to create the database for the correlation analysis (see subsection 5.3.1). Second, all models analysed and the structural metrics (according to Table 5-1) are computed (see subsection 5.3.2). Third, the correlations among the metrics are computed and documented in correlation matrices (see subsection 5.3.3). Fourth, the correlation matrices are clustered and the metric of the minimal set sets are derived (see subsection 5.3.4). Fifth, the proposed minimal set is validated in two case studies by showing the significance of the analysis and that the set provides the same insights as all available metrics (see subsection 5.3.5).

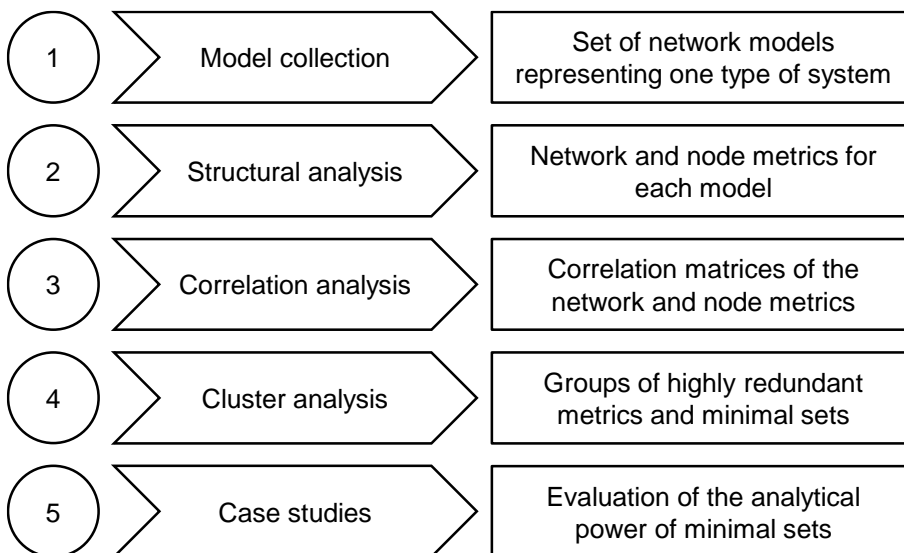


Figure 5-2 Main activities and their outcome for determining and evaluating the metric of the minimal set

5.3.1 Research for model collections

Two model collections were researched: a collection from a literature review and a collection from a design repository. The literature collection shall cover a wide range component networks models and is the database for finding a candidate set of metrics of the minimal set. The repository collection serves the testing of the candidate set.

First, literature was reviewed to create a collection of structural product models. The creation of the literature-based collection is also described in [BIEDERMANN & LINDEMANN 2012]. The collection should cover a wide range in terms of the model type (e.g. physical and functional), model size and product type (e.g. purely mechanical and mechatronic). Seventeen major (based on the 2010 ISI rating) journals and the volumes of the past ten years were reviewed. Some volumes were not available due to licencing. Five journals are from the area of engineering design, five from engineering management and seven from systems engineering. Decision science and operations management were omitted as this thesis focuses on product models. The review resulted in about 100 publications providing models of product structures. These publications were reduced to publications, which provide models fulfilling following requirements:

- Models describe a physical product
- Models are intended to be used in product design and development
- Models are fully available and not anonymised
- Models describe components and their interrelations
- Models describe undirected relations

After the reduction twenty models remained. The collection was supplemented by models available at the institute, which have not been published. The final collection comprises 35 models. The literature models were extracted from the publications and transformed into a text-based, computable matrix format. Subsection 6.1.1 gives an overview of the collection and its characteristics in terms of type of relations, type of system, model size and data acquisition method. Section 11.5 lists all models of the collection, their characteristics and the original references.

Second, a collection of models was derived from the Design Repository hosted by the Design Engineering Lab of the Oregon State University [BOHM ET AL. 2005]. The repository provides models of natural and technical system. The available views also include DSMs. The structural models were downloaded on March 4th 2012 and transformed into a text-based, computable matrix format. Originally the collection comprised more than 140 models. However, some models were not used due to one the following reasons:

- The model describes an animal
- The model contains no relations
- The model is too large (i.e. the metrics could not be computed with 24 hours)

The remaining set contains 124 models. Subsection 6.1.2 gives an overview of the collection and its size characteristics. Section 11.6 lists all models of the collection and their characteristics. The repository does not define the type of elements or the type of relations in the models. Based on the general description of the repository, the insights of chapter 2 and the

content of the models (e.g. nomenclature of the elements and level of detail) the assumption is that the models describe parts of hardware products, which are related via contacts.

5.3.2 Computations of the structural metrics

Next, the structural metrics were computed using LOOMEOTM 2.5.0 [TESEON 2011] (four global and six local metrics) and macros for Microsoft ExcelTM 2010 (nine global and four local metrics). The computed metrics and the computation tool are listed in Table 5-1 and described in the appendix (sections 11.2 and 11.3). The computation resulted in one text file for each model listing the nodes and the metric values. Two additional files list the global metric values for each collection. The results of the structural analysis are not listed in this thesis for the sake of brevity.

5.3.3 Correlation analysis of structural metrics

The dependencies among the metrics were identified using a correlation analysis. The computation of the matrices is also described in [BIEDERMANN & LINDEMANN 2012A]. The dependencies are the basis for the grouping of the metrics. For each dataset (file) a correlation matrix of the metrics and the level of significance for each correlation was computed. Each matrix cell contains the Pearson coefficient [STEIGER 1980] of the two connected metrics. The level of significance [CHOW 1996] was computed using the Student's t-test [BONEAU 1960]. For each model collection the correlation matrices of the local metrics were averaged. All statistical analyses were done with Microsoft ExcelTM 2010. The resulting correlation matrices are not presented in this thesis as they are reordered in the next step to reveal clusters among them. The clustered matrices are presented in section 6.2.

5.3.4 Determination of metrics of the minimal set sets

Last, the metrics were grouped based on the dependencies among them. For each group one metric was determined, which can represent the group as it is highly correlated to the other group metrics.

To find the groups the correlation matrices of the literature collection were clustered to highlight sets of highly correlated metrics. The clustering of the matrices is also described in [BIEDERMANN & LINDEMANN 2012]. The clustering was done manually due to the small matrix sizes (13 by 13 and 10 by 10) – see also section 3.4 for an introduction to matrix clustering. Each cluster represents one group of highly interdependent metrics. Due to the limitations of manual clustering the groups may overlap or the assignment of one metric to a group may be arbitrary. The resulting matrices are shown in section 6.2. Figure 6-1 shows the correlation matrix of the global metrics for the literature collection. Figure 6-3 shows the averaged correlation matrix of the local metrics for the literature collection.

The representation candidate for each group was chosen based on the highest minimal Pearson coefficient within the group. Some groups have more than one candidate. One group comprises also rather low coefficient values (see section 6.2). Therefore, alternative candidates and

supplementary metrics were proposed. The candidates for the minimal set are presented in section 6.3.

The correlations matrices of the repository collection were not clustered. Rather they were reordered to the same sequence of metrics as the matrices of the literature sets. Thereby, the findings of the literature set can be tested (and to a certain degree validated). If matrices of the repository collection show the same patterns as the matrices of the literature collection the findings are confirmed. Deviations may lead to an update of the metrics groups and representatives. The matrices are shown in Figure 6-2 (global metrics) and Figure 6-4 (local metrics). The deviations are discussed in section 6.2 and the consequences for the minimal sets are presented in section 6.3.

5.3.5 Validation of metric of the minimal set sets

Whereas the previous subsections describe the testing of the disparity criterion (see subsection 4.1.2) this section focuses on the validation via testing the significance (see subsection 4.1.3 and section 4.2) of the minimal set. The minimal set must provide the same analytical power as all metrics together (see section 4.4). Otherwise the set is not valid and must be changed, extended or dismissed. This thesis provides two case studies to test the validity of the minimal set. The case studies have to fulfil four main requirements:

- Each study must use structural analyses
- Each study must address a relevant problem in engineering design
- Each study must show the significance of the structural analyses
- Each study must show that the minimal set provide all relevant insights

The first case study (see chapter 7) analyses how structural characteristics impact the change simulation results. The study uses a simulation approach (see subsection 4.2.3) to show the significance of the results. It has already been partially published in [BIEDERMANN ET AL. 2010], [BIEDERMANN ET AL. 2011] and [BIEDERMANN ET AL. 2012]. The second case study (see chapter 8) addresses the assessment of the reconfigurability of production resources. The study uses an empirical approach (see subsection 8) to show the significance of the results. It has already been partially published in [BIEDERMANN ET AL. 2010], [ZAEH ET AL. 2010] and [ZAEH ET AL. 2011]. The detailed approaches, methodology and results of each case study are provided in the chapters 7 and 8.

5.4 Summary

This chapter describes the research methodology to identify and validate sets of metrics of the minimal set. By reducing the number of metrics, structural analysis becomes easier to learn, to apply and to research.

The first section (5.1) provides the theoretical background for identifying minimal sets. Metrics (and structural analyses in general) may be redundant if they depend on or relate to each other. Therefore, relations among the metrics allow for eliminating metrics. Some relations result from the definition of the metrics. Other relations are characteristics for the class of models. Only the latter allow for identifying minimal sets as they are directly linked to the redundancy

among metrics. In particular highly correlated metrics are mostly redundant. Therefore, this thesis focuses on determining the correlations among the metrics for a class of models.

The second section (5.2) lists the metrics to be researched. They were derived from the state of the art, network theory and graph theory. So far not all available metrics have been addressed in research on structural analysis. Yet, thesis includes them a well to assess their potential for future research. The final list comprises ten local and thirteen global metrics.

The third section (5.3) describes the actual research methods of this thesis: starting with collecting the required models, going over the computation and correlation analysis of the metric values and ending with the identification of the minimal sets of global and local metrics. The main method for testing the disparity among the metrics is correlation analysis of the values. The correlated metrics are then clustered to identify highly redundant groups of metrics, which are then reduced to one metric among them. Once the metric sets are known they still have to be validated. The main method for validating the proposed minimal set is showing its feasibility in case studies. Two case studies are presented in this thesis.

The next chapter presents the results of the research methods including the model collections, the results of the correlation analysis and the minimal sets of global and local metrics for the analysis of component networks. The chapters 7 and 8 present the case studies for validating the metric of the minimal set sets.

6. Metric of the minimal set for concept analysis

Chapter 4 introduces the main approach of this thesis: finding minimal sets of structural metrics for product component networks. Chapter 5 describes the research methodology for finding and validating the metric of the minimal set sets. This chapter presents the results of the research in particular the metric of the minimal set sets. However, the validation is only partially presented here. The actual validation is presented in the case study chapters 7 and 8.

The definition of the metric of the minimal set sets bases on a correlation analysis of several product models (see section 5.3 for details). Section 6.1 presents the model collections resulting from a literature review and a design repository (see subsection 5.3.1). There are two sets of component networks: one for identifying the metric of the minimal set sets and one for validating the sets. The first model collection (from the literature review) comprises 35 component networks without focus a particular type of relation (see 6.1.1); the second set (from the repository) comprises 124 component networks with contact relations (see 6.1.2).

Section 6.2 presents the results of the correlations analysis (see subsections 5.3.2 to 5.3.4 for the methodology). The correlation analysis confirms that the metrics highly interdepend and form several clusters. Based on the clusters and the comparison of the results for the two collection two minimal sets are proposed: one for global metrics and one for local metrics.

Section 6.3 discusses the two sets and their characteristics in detail. The minimal set for global metrics, i.e. metrics analysing entire products, comprises six metrics. Future research might reduce the set further due to new findings. The set for local metrics, i.e. metrics analysing single components in a network, comprises four metrics. The comparison of the collections partially confirms the findings of the correlation analysis. However, the characteristics of two global metrics and one local metric differ for the collections.

6.1 Model collections

There are two collections of component networks: one for identifying the metric of the minimal set and one for validating the set. The first model collection comprises 35 component networks without focus a particular type of relation and was derived in an extensive literature review (see subsection 5.3.1). The models in the first set cover a wide range of sizes, types of relations, way of creation and types of systems. The second set comprises 124 components networks with contact relations and was taken from an online repository of product designs. The second set focuses on one type of relations and one way of creation but covers a wide range of sizes and systems.

6.1.1 Literature review

The collection of the data and the properties of the models have already been described in [BIEDERMANN & LINDEMANN 2012A]. In the literature review 35 models, which describe 23 products, were found. Section 11.5 of the appendix shows the models with additional data like

the relationship type, the way of data acquisition, the reference, the number of nodes, the number of edges and the original resources. The models from the references [EINÖGG 2009], [LANGER ET AL. 2010], [MAURER 2011], [SCHMITZ ET AL. 2011], [STRELKOW 2010] and [TESEON 2011] are not publicly available as they:

- have not been published so far ([EINÖGG 2009] and [STRELKOW 2010])
- are part of course work ([MAURER 2011])
- show not all available data ([LANGER ET AL. 2010], [SCHMITZ ET AL. 2011])
- are part of a software release ([TESEON 2011])

The classification of the relationship type follows the proposition by [PIMMLER & EPPINGER 1994] and the findings of section 2.1:

- Product (4 models): no specification beyond component structure of a product
- Geometry (8 models): geometric links such as contact or design space intersection
- Contact (13 models): specific geometric link – the components are in contact
- Flow (5 models): flows between the components such as energy, information or heat
- Function (5 models): relations linked to the overall system functionality

The classification scheme for data acquisition was newly created for this thesis. Three ways are distinguished (the list is by no means complete; other ways such as questionnaires and data mining are not used in the references)

- Work on product (19 models): the modeller has access to the product and can disassemble it
- Interview/workshop (6 models): the modeller interviews product experts to create the model
- No mention (10 models): the reference does not state the mode of the data acquisition

The models span a wide range of sizes. The number of nodes runs from 7 to 110 with an average around 30. The number of edges runs from 9 to 147 with an average around 40. The type of systems varies from purely mechanical products such as ball-pens and sprinklers to highly integrated mechatronic products such as cell phones or assembly cells. Thus, the models cover a wide range of products and model types.

6.1.2 Design Repository

Section 11.6 lists all 124 models of the collection and their characteristics. The repository does not define the type of elements or the type of relations in the models. Based on the general description of the repository, the insights of chapter 2 and the content of the models (e.g. nomenclature of the elements and level of detail) the assumption is that the models describe parts of hardware products, which are related via contacts. Moreover, no information concerning the modelling process is provided.

The models span a wide range of sizes. The number of nodes runs from 8 to 114 with an average around 35. The number of edges runs from 1 to 116 with an average around 44. 18 models contain more elements than relations. The lack of relations results from the type of systems: many toys contain elements, which are not fixed to the main product. The product in the collection fall into four main categories: toys (e.g. a ball shooter), household appliances (e.g.

coffee machines), tools (e.g. drilling machines) and electronic devices (e.g. cell phones). Thus, the collection covers a wide range of products and sizes.

6.2 Correlation matrices of the metrics

This section presents the results of the correlation analyses. The correlation analysis confirms that the metrics highly interdepend and form clusters. Subsections 5.3.2 to 5.3.4 describe the creation of the clustered correlation matrices, which are shown in Figure 6-1 to Figure 6-4. The intermediate results are omitted for the sake of brevity.

Many of the correlations among the global metrics are significant for both model collections (see subsection 6.2.1). The original cluster analysis for the literature collection indicates five metrics for the minimal set. However, the correlation matrices of the two collections differ substantially. Therefore, the proposed set is updated to comprise six metrics.

The correlations among the local metrics (see subsection 6.2.2) form very clear clusters. They indicate a minimal set of four metrics. However, the deviations among the two collections suggest a further reduction to three metrics, which has to be tested in future research.

6.2.1 Correlations for global metrics

Figure 6-1 shows the clustered correlation matrix of the global metrics for the literature collection. The matrix contains eleven significant ($p < 0.05$), five very significant ($p < 0.01$) and twenty-three highly significant ($p < 0.001$) correlations. The matrix shows four clusters.

The first cluster contains three metrics: “average clustering coefficient”, “average degree” and “average number of cliques per node”. All correlations within the cluster are highly significant. The metric “average number of cliques per node” has the highest correlations to the other metrics (0.83 and 0.93). Thus, it is the candidate for representing the complete cluster. Both other metrics may be suitable as well. As the “average degree” can be computed far more efficiently it is suggested as metric of the minimal set.

The second cluster contains eight metrics: “number of cliques”, “average path length”, “average distance centrality”, “number of edges”, “average path centrality”, “number of nodes”, “number of blocks” and “number of components”. Seventeen correlations within the cluster are highly significant; four correlations are significant and five correlations are not significant. Thus, the cluster is not as clear cut as the first cluster and requires careful treatment. The metric “number of edges” has the highest correlations to the other metrics (ranging from 0.25 to 0.95). Thus, it is a candidate for representing the entire cluster. As the cluster is not completely significant other metrics have to supplement the “number of edges”. The prime candidate is “number of nodes” due to its highly significant correlations to the rest.

The third cluster contains only the metric “Average number of blocks per node”, which can represent itself and the cluster.

The fourth cluster contains two metrics: “average number of cycles per node” and “number of cycles”. The correlation within the cluster is highly significant. Therefore, both metrics can

represent the cluster. The “number of cycles” is proposed as the candidate as its computation requires slightly less effort.

The first draft for the minimal set of global metrics is:

- average degree
- average number of blocks per node
- number of cycles
- number of edges
- number of nodes

	Average clustering coefficient	Average degree	Average number of cliques per node	Number of cliques	Average path length	Average distance centrality	Number of edges	Average path centrality	Number of nodes	Number of blocks	Number of components	Average number of blocks per node	Average number of cycles per node	Number of cycles
Average clustering coefficient		0,74	0,83	0,41	0,09	0,12	0,02	-0,26	-0,36	-0,47	-0,47	-0,03	0,01	0,06
Average degree	0,74		0,93	0,60	0,14	0,37	0,28	-0,16	-0,22	-0,47	-0,46	-0,34	0,35	0,47
Average number of cliques per node	0,83	0,93		0,58	0,03	0,27	0,21	-0,21	-0,22	-0,41	-0,36	-0,27	0,23	0,33
Number of cliques	0,41	0,60	0,58		0,59	0,86	0,78	0,40	0,36	0,09	-0,17	-0,10	0,29	0,32
Average path length	0,09	0,14	0,03	0,59		0,71	0,59	0,60	0,34	0,21	-0,27	0,33	0,00	0,00
Average distance centrality	0,12	0,37	0,27	0,86	0,71		0,95	0,71	0,63	0,39	-0,03	-0,02	0,24	0,27
Number of edges	0,02	0,28	0,21	0,78	0,59	0,95		0,80	0,81	0,58	0,25	-0,15	0,23	0,26
Average path centrality	-0,26	-0,16	-0,21	0,40	0,60	0,71	0,80		0,85	0,81	0,33	0,10	-0,06	-0,07
Number of nodes	-0,36	-0,22	-0,22	0,36	0,34	0,63	0,81	0,85		0,92	0,72	-0,14	0,02	0,03
Number of blocks	-0,47	-0,47	-0,41	0,09	0,21	0,39	0,58	0,81	0,92		0,77	0,09	-0,13	-0,13
Number of components	-0,47	-0,46	-0,36	-0,17	-0,27	-0,03	0,25	0,33	0,72	0,77		-0,29	-0,06	-0,07
Average number of blocks per node	-0,03	-0,34	-0,27	-0,10	0,33	-0,02	-0,15	0,10	-0,14	0,09	-0,29		-0,21	-0,23
Average number of cycles per node	0,01	0,35	0,23	0,29	0,00	0,24	0,23	-0,06	0,02	-0,13	-0,06	-0,21		0,99
Number of cycles	0,06	0,47	0,33	0,32	0,00	0,27	0,26	-0,07	0,03	-0,13	-0,07	-0,23	0,99	

Level of significance:

significant (p < 0.05): 0.33 < |values| < 0.42

very significant (p < 0.01): 0.45 < |values| < 0.48

highly significant (p < 0.001): 0.57 < |values|

Figure 6-1 Correlation matrix of the global metrics for the literature model collection

Figure 6-2 shows the clustered correlation matrix of the global metrics for the repository collection. The matrix contains ten significant ($p < 0.05$), five very significant ($p < 0.01$) and fifty highly significant ($p < 0.001$) correlations. The increase of the level of significance results from the higher number of models in the data base.

	Average clustering coefficient	Average degree	Average number of cliques per node	Number of cliques	Average path length	Average distance centrality	Number of edges	Average path centrality	Number of nodes	Number of blocks	Number of components	Average number of blocks per node	Average number of cycles per node	Number of cycles
Average clustering coefficient		0,75	0,91	0,67	0,27	0,40	0,30	-0,04	-0,16	-0,36	-0,40	0,18	0,15	0,17
Average degree	0,75		0,86	0,72	0,59	0,66	0,53	0,13	-0,10	-0,39	-0,60	0,41	0,24	0,23
Average number of cliques per node	0,91	0,86		0,77	0,33	0,49	0,39	-0,04	-0,11	-0,38	-0,40	0,18	0,26	0,19
Number of cliques	0,67	0,72	0,77		0,38	0,81	0,83	0,34	0,47	-0,07	-0,22	0,11	0,36	0,58
Average path length	0,27	0,59	0,33	0,38		0,59	0,45	0,69	0,08	0,01	-0,51	0,69	0,09	0,12
Average distance centrality	0,40	0,66	0,49	0,81	0,59		0,94	0,61	0,52	0,19	-0,21	0,32	0,19	0,47
Number of edges	0,30	0,53	0,39	0,83	0,45	0,94		0,65	0,67	0,27	-0,07	0,17	0,26	0,57
Average path centrality	-0,04	0,13	-0,04	0,34	0,69	0,61	0,65		0,59	0,51	-0,07	0,39	0,03	0,25
Number of nodes	-0,16	-0,10	-0,11	0,47	0,08	0,52	0,67	0,59		0,65	0,61	-0,19	0,08	0,32
Number of blocks	-0,36	-0,39	-0,38	-0,07	0,01	0,19	0,27	0,51	0,65		0,55	0,19	-0,10	0,02
Number of components	-0,40	-0,60	-0,40	-0,22	-0,51	-0,21	-0,07	-0,07	0,61	0,55		-0,59	-0,06	-0,02
Average number of blocks per node	0,18	0,41	0,18	0,11	0,69	0,32	0,17	0,39	-0,19	0,19	-0,59		-0,03	-0,01
Average number of cycles per node	0,15	0,24	0,26	0,36	0,09	0,19	0,26	0,03	0,08	-0,10	-0,06	-0,03		0,98
Number of cycles	0,17	0,23	0,19	0,58	0,12	0,47	0,57	0,25	0,32	0,02	-0,02	-0,01	0,98	

Level of significance:

significant ($p < 0.05$): $0.17 < |\text{values}| < 0.24$

very significant ($p < 0.01$): $0.23 < |\text{values}| < 0.28$

highly significant ($p < 0.001$): $0.29 < |\text{values}|$

Figure 6-2 Correlation matrix of the global metrics for the repository model collection

The second correlation matrix is less clearly structured than the first one. The first and fourth clusters are visible in matrix but the second and third cluster are much less pronounced. In

particular the metrics “number of cliques” and “average number of blocks per node” differ. “Number of cliques” is part of the second cluster in the literature matrix. In the repository matrix it rather is connected to the first cluster. “Average number of blocks per node” forms an individual cluster in the first matrix. In the second matrix it is connected to the second clusters. Moreover, there are more significant relations between the clusters in the second matrix. In particular the second cluster is more related to the first and fourth cluster. The representing metrics of the second cluster (“number of edges” and “number of nodes”) are highly correlated to other metrics in both matrices.

The differences between the matrices call the first draft of the minimal set into question. There are two ways to handle this. First, the second matrix can be clustered and a second proposal for the minimal set can be derived. The two proposals are then to be tested with additional data. This option is dismissed due to the lack of additional testing data. Second, the minimal set is extended to include the metrics in question. Future research has to show the correct set. Thus, the final draft for the minimal set of global metrics is:

- average degree
- average number of blocks per node
- number of cliques
- number of cycles
- number of edges
- number of nodes

Due to the observations not all metrics are indisputable. Two metrics are confirmed by both matrices as parts of the minimal set: “average degree” and “number of cycles”. Two metrics are disputable candidates in both matrices: “number of edges” and “number of nodes”. Two metrics show different characteristics in the two matrices: “average number of blocks per node” and “number of cliques”.

6.2.2 Correlations for local metrics

Figure 6-3 shows the clustered correlation matrix of the local metrics for the literature collection. The structure of the correlations is clearer than for the global metrics. There are four clusters in the matrix.

The first cluster contains five metrics: “degree”, “path centrality”, “cycles per node”, “blocks per node” and “cliques per node”. The metric “degree” has the highest correlation values to the other metrics (ranging from 0.63 to 0.84). Thus, it is part of the first proposal for the minimal set.

The second and third clusters contain only one metric each: “clustering coefficient” and “distance centrality”. Both metrics are part of the first proposal for the minimal set.

The fourth cluster contains three metrics: “average distance to node”, “maximum distance to node” and “median distance to node”. The metric “average distance to node” has the highest correlations to the other metrics (0.86 and 0.91). Thus, it is a candidate for representing the complete cluster.

The first draft for the minimal set of global metrics is:

- degree
- clustering coefficient
- distance centrality
- average distance to node

	Degree	Path centrality	Cycles per node	Blocks per node	Cliques per node	Clustering coefficient	Distance centrality	Average distance to node	Maximum distance to node	Median distance to node
Degree		0,80	0,84	0,63	0,83	-0,04	0,12	-0,08	-0,02	-0,10
Path centrality	0,80		0,52	0,80	0,55	-0,25	0,15	-0,14	-0,09	-0,14
Cycles per node	0,84	0,52		0,34	0,71	0,05	0,09	-0,05	0,00	-0,07
Blocks per node	0,63	0,80	0,34		0,39	-0,18	0,06	-0,05	-0,03	-0,07
Cliques per node	0,83	0,55	0,71	0,39		0,21	0,09	-0,07	-0,01	-0,07
Clustering coefficient	-0,04	-0,25	0,05	-0,18	0,21		-0,09	0,11	0,09	0,13
Distance centrality	0,12	0,15	0,09	0,06	0,09	-0,09		-0,49	-0,31	-0,44
Average distance to node	-0,08	-0,14	-0,05	-0,05	-0,07	0,11	-0,49		0,86	0,91
Maximum distance to node	-0,02	-0,09	0,00	-0,03	-0,01	0,09	-0,31	0,86		0,74
Median distance to node	-0,10	-0,14	-0,07	-0,07	-0,07	0,13	-0,44	0,91	0,74	

Figure 6-3 Correlation matrix of the local metrics for the literature model collection

Figure 6-4 shows the clustered correlation matrix of the local metrics for the repository collection. The correlation values have about the same values (in terms of the order of magnitude) as for the literature collection. Moreover, the matrix has the same structure as the first with one exception. The first, second and fourth cluster are also present in the matrix. Therefore, the corresponding metrics for the minimal set are confirmed.

The main difference between the matrices is the metric “distance centrality”. For the literature collection it forms a singular cluster and is therefore part of the minimal set proposal. For the repository collection it seems to be part of the first cluster and can thus be presented by “degree” as the other metrics of the cluster. Another striking difference is the correlation to the fourth cluster.

The comparison of the two matrices confirms three matrices of the minimal set: “degree”, “clustering coefficient” and “average distance to node”. The fourth metric (“distance

centrality”) must be reassessed in future research. Therefore, it is still part of the minimal set proposal for local metrics.

	Degree	Path centrality	Cycles per node	Blocks per node	Cliques per node	Clustering coefficient	Distance centrality	Average distance to node	Maximum distance to node	Median distance to node
Degree		0,81	0,85	0,65	0,78	0,11	0,79	0,03	0,14	-0,03
Path centrality	0,81		0,56	0,65	0,56	-0,11	0,54	-0,12	-0,06	-0,14
Cycles per node	0,85	0,56		0,42	0,67	0,15	0,73	0,03	0,13	-0,03
Blocks per node	0,65	0,65	0,42		0,33	0,02	0,62	0,30	0,35	0,26
Cliques per node	0,78	0,56	0,67	0,33		0,41	0,58	-0,10	-0,01	-0,13
Clustering coefficient	0,11	-0,11	0,15	0,02	0,41		0,23	0,12	0,15	0,08
Distance centrality	0,79	0,54	0,73	0,62	0,58	0,23		0,38	0,48	0,29
Average distance to node	0,03	-0,12	0,03	0,30	-0,10	0,12	0,38		0,93	0,94
Maximum distance to node	0,14	-0,06	0,13	0,35	-0,01	0,15	0,48	0,93		0,83
Median distance to node	-0,03	-0,14	-0,03	0,26	-0,13	0,08	0,29	0,94	0,83	

Figure 6-4 Correlation matrix of the local metrics for the repository model collection

6.3 Metric of the minimal set sets for local and global analysis

Section 6.2 shows that structural metrics are highly correlated and that the global metrics are highly significantly correlated for component networks. The metrics form highly correlated clusters. For each cluster candidates for representing the whole cluster are proposed. This shows that the set of sensibly applicable structural analyses can be reduced based on the type of system.

The results suggest that whole product structures can be characterised by a set of six metrics. The set should be sufficient to gain an overview of the structure and to gain some fundamental insights. As some of the candidates are hard to determine or seem not sufficient alternatives and supplements are proposed. The final proposal contains:

- average degree
- average number of blocks per node
- number of cliques

- number of cycles
- number of edges
- number of nodes

“Number of edges” and “number of nodes” are commonly used to basically characterise graphs [HARARY ET AL. 1965] and networks [CAMI & DEO 2008]. However, usually they merely give a structural overview but yield no further implications. Future research has to show if they allow for further analyses e.g. assessing life-cycle properties or product inherent properties. Interestingly, both metrics are hard correlated to the “average degree” even though the “average degree” is the ratio between “number of edges” and “number of nodes” [HARARY ET AL. 1965]. Therefore, the three metrics seems to be independent and to allow for separate analyses. “Average number of blocks per node”, “number of cliques” and “number of cycles” have hardly been used in structural analysis. Their implications have to be assessed in further research. Interestingly, the “average path length” is not part of the minimal set even though it is a metric of the minimal set in network theory to classify networks [CAMI & DEO 2008].

The results suggest that nodes can be characterised by a set of four metrics. The set should be sufficient to gain an overview and some fundamental insights. However, these findings must be confirmed by an additional analysis as the results are not significant. The final proposal contains:

- degree
- clustering coefficient
- distance centrality
- average distance to node

“Degree” and “clustering coefficient” are common metrics in network theory and a wide body of research addressed their characteristics in natural and large-scale technical networks [Cami & Deo 2008]. The “degree” is also commonly used in structural analysis of components networks (see chapter 3). The two metrics supplement each other: “degree” is the number of neighbouring elements, “clustering coefficient” measures the cross-linking among the neighbours. The two distance metrics characterise the global embedding of the element in the network.

6.4 Conclusion

Based on the analysis of two model collections two metric of the minimal set sets were identified: one for global metrics and one for local metrics. The two model collections cover about the same range of model sizes (both in terms of elements and relations) but differ in total number, variety of type of relations and the focus of the types of systems.

The global metric set is ambiguous. Two metrics (“average number of blocks per node” and “number of cliques”) may not be required in the minimal set. Based on the available data no further research is sensible. Future research has to determine the role of the two metrics in the minimal sets. However, the analyses of the two collections contradict each other for the two metrics. Therefore, both should be tested in future research. Moreover, the remaining four metrics are not entirely clear as well. Whereas, “average degree” and “number of cycles” are confirmed by both analyses, the other two metrics (“number of nodes” and “number of edges”)

are not clear candidates. Both are part of the same cluster, which also contains lowly correlated metrics. They are primarily suggested as they cover the entire cluster together and are very easy to determine. Thus, the minimal set of global metrics has to be carefully assessed in future research. So far, it comprises six metrics:

- average degree
- average number of blocks per node
- number of cliques
- number of cycles
- number of edges
- number of nodes

The local metric set is unambiguous. Three of the four metrics are confirmed in the second analysis. The fourth metrics “distance centrality” may be discarded by future research. It should be used in structural analysis. However, it must always be tested if it can be replaced by the metric “degree” as the analysis of the repository collection suggests. So far, the minimal set of local metrics comprises the four metrics:

- degree
- clustering coefficient
- distance centrality
- average distance to node

By identifying the minimal sets the requirement for disparity among the metrics is proved for component network models of hardware products. The minimal sets need to be tested in real applications to show that they really provide the same analytical power as all metrics. The next two chapters (7 and 8) provide case studies for validating the minimal set of local metrics.

7. Simulation analysis of product change dynamics

The previous chapter proposes two minimal sets of structural metrics. This chapter validates the set for local metrics using a simulation approach. The idea is to show that the minimal set provides the same insights as all available metrics. Subsection 5.3.5 discusses the validation. There are two levels of validation. The lower level simply means that the other metrics provide no new insights. The higher level means that the metrics of the minimal set provide the same insights as the corresponding cluster (see chapter 6 for the cluster definitions).

The case study has to fulfil four main requirements according to subsection 5.3.5: application of structural analyses, relevant problem in engineering, proof of the significance of the structural analysis and proof of the validity of the minimal set.

This case study analyses, how structural characteristics impact change simulation results. Chapter 3 provides an introduction to two change-related topics: changeability (subsection 3.2.2) and change propagation (section 3.5). In particular, the empirical work on change propagation has shown the relevance of change prediction – both structure-based and simulation-based. Due to the high effort for creating simulation models it must be shown that they provide additional insights. Delimiting the analytical power of structure-based and simulation-based approaches is the key aim of the case study.

The study uses a simulation approach (see subsection 4.2.3) to show the significance of the results. However, the approach substantially differs from previous work. The simulation model bases on cellular automata (see [WOLFRAM 1983]) rather than probabilistic models. Moreover, the simulation runs cover the entire parameter space of the simulation model rather than using scenarios or Monte-Carlo-approaches. Nevertheless the resulting data only allows for qualitative instead of statistical analyses for showing the significance of the results. The same applies for the proof of validity for the minimal set.

This case study has already been partially published in [BIEDERMANN ET AL. 2010], [BIEDERMANN ET AL. 2011] and [BIEDERMANN ET AL. 2012]. The structural modelling and the implementation of the simulation are also described in the theses [STRELKOW 2010] and [LÜNING 2011], which are not publicly available. Moreover, the study uses preliminary work by [DIEPOLD ET AL. 2010A] and [DIEPOLD ET AL. 2010B]. To keep the presentation brief and focused many details of the simulation model and the simulation set ups are omitted in the text but are provided in section 11.7 of the appendix. Moreover, the publications above are referred to if appropriate or necessary.

7.1 Simulation of product changes

One of the major challenges in engineering management is the ability to respond quickly to new and/or changed requirements and constraints. To establish, ensure and improve this ability several measures exist for the design of products (e.g. modularisation), processes (e.g. agile development) and organisations (e.g. task forces). Models of the products, processes and organisations are fundamental for their design [LINDEMANN ET AL. 2009].

There are two major modelling approaches for change prediction of engineering systems: structure-based (e.g. design structure matrices – DSM) and dynamic-based (e.g. differential equations or fuzzy systems) models. There are also some mixed forms, e.g. Petri-nets. Dynamic-based models generally allow for precise analyses. They are usually used for detailed planning and optimisation. The major drawback of dynamic-based models is the need for a lot of data (or estimations if no data is available) when creating the models. Hence, dynamic-based models usually describe only small parts or single effects of engineering systems [DIEPOLD ET AL. 2010A]. Structure-based models allow for general analyses. They are mostly used for early planning and system decomposition. Compared to dynamic-based models structure-based models require rather little data. Most structure-based models claim to describe the engineering system completely [BROWNING 2001]. To sum it up: structure-based approaches show potential changes, simulation-based approaches plan and avoid changes.

To improve the change management of engineering systems the strengths of both approaches ought to be combined while avoiding their drawbacks. Understanding the interconnection between structure and dynamic of a system is the major key to handle a system successfully [STROGATZ 2001].

[DIEPOLD ET AL. 2010A] introduce the combination of structural analysis and modelling of the system dynamics. The major interfaces between both concepts are substructures of the system, which mainly determine its behaviour (see also [KREIMEYER & LINDEMANN 2011]). Structural analysis provides methods for identifying those substructures based on structural patterns such as cycles. [DIEPOLD ET AL. 2010A] introduce a process to combine both methods and transform the results of structural analysis into a simulation model. Figure 7-1 shows the process and its major artefacts.

A more detailed framework for transforming a structural model into a simulation model is introduced in [DIEPOLD ET AL. 2010B]. The framework consists of structural analysis, qualitative error estimation, dynamical modelling, error estimation and data acquisition for refinement (if required). However, they do not discuss which structural analyses are suitable for identifying refinement potential and for deducing dynamical behaviour.

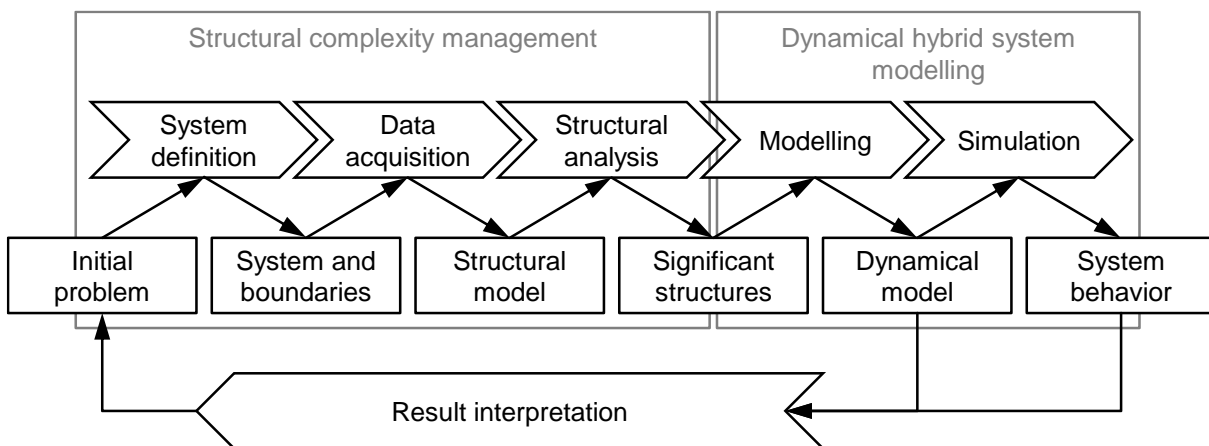


Figure 7-1 Modelling process for multi-dynamic mapping

This chapter compares the behaviour of system components with its structural properties. The case study uses a component model of an assembly cell [BIEDERMANN ET AL. 2010]. The behaviour is determined via simulation. The simulation describes the system response when changing the size of a component.

7.2 Simulation runs and assessment

The simulation is summarised in the following (a detailed description of the simulation approach is given in [BIEDERMANN ET AL. 2011]). The simulation model fulfils three major tasks. First, the required component changes are detected based on its current state and violated consistence conditions with related components. Second, deciding how a component should be finally adapted in the next time step. The algorithm gathers requests for changing a certain component and commands its new state value by taking pre-defined prioritizing policies and change options into account. Third, it allow for evaluating the simulation results.

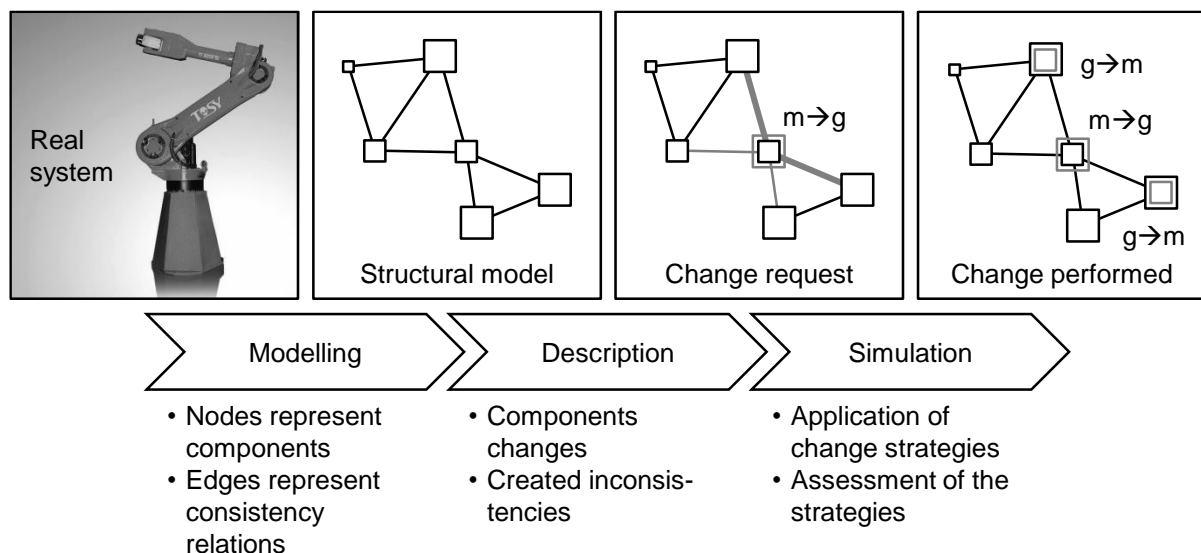


Figure 7-2 Decision-based change simulation

(robot image taken from http://commons.wikimedia.org/wiki/File:TOSY_Arm_Robot1.jpg)

This case study does not use the standard simulation model for change prediction (see section 3.5 for a description of the standard model). The standard model uses a probabilistic approach for modelling knock-on changes. However, this requires in depth knowledge and experience about the systems. Most studies use only the most likely scenarios or a Monte-Carlo-simulation to predict changes. This also requires high familiarity with the systems. When designing new systems often not enough knowledge exists. Therefore, this case study directly simulates the change decisions. Namely, how are the components to be changed and which components are to be changed next. Both decisions are guided by simple policies (see appendix 11.7) based on violated consistency relations between components. To cover all potential changes the case study runs simulations for the entire parameter space of the simulation model. However, the

case study makes several simplifications and assumptions: there is only one initial change, each component modification takes the same time and effort and there is no change planning.

The example product for the case study is an assembly cell. The structural model of the assembly cell is a DSM of their components' physical contacts. The model comprises 110 nodes and 147 relations. The time-discrete simulation model is based on a network topology, where nodes represent components and edges are dependencies between them, and it is computationally handled as cellular automaton [WOLFRAM 1983]. In order to allow for a simulation, each component is attributed with its volume as state variable. A discrete scale ("small", "medium", "large") is introduced and all components are matched to the scale. The necessity of a change is derived by evaluating consistence constraints. For instance, component A must be larger or equal to component B. The considered consistence relations are summarised in subsection 11.7 of the appendix.

Each simulation is initialised by changing the size of one component to create the initial change situation. After that the simulation runs according to the policies and the consistence relations. For each simulation run the number of size changes, the number of simulated time steps and the type of the final result (see also appendix 11.7) were recorded. To compare the results with structural properties, two simulation metrics were computed for each component (each representing 6468 simulation runs):

- Standard deviation of the number of changed elements: During each simulation several components are changed according to the change policies and the consistence relations. The standard deviation measures the degree of variation of knock-one changes. If a component has a low standard deviation its behaviour hardly varies and is very predictable.
- Standard deviation of the number of time-steps: Each simulation covers several time steps until the states of components are not changed any more. The standard deviation measures the degree of variation of the simulation time. If a component has a low standard deviation its behaviour hardly varies and is very predictable. This research examines if such components can be identified by structural analysis.

The components of the assembly cell were structural characterised by the four metrics of the minimal set: degree, clustering coefficient, distance centrality and average distance to node. Additionally, six metrics were computed to allow for testing the validity of the minimal set: blocks per node, cliques per node, cycles per node, maximum distance to node, median distance to node and path centrality. The metrics were computed with LOOMEOTM 2.5.0 and Microsoft ExcelTM 2010 (see also Table 5-1).

To examine the interrelation between structural and simulation metrics scatter plots were created. Figure 7-4 shows the scatter plots for the four local metrics of the minimal set. Figure 7-5 compares the degree metric to its potential substitutes. Due to data limitations further statistical analyses were relinquished such as analyses of means, correlations and significance.

7.3 Structural properties of changing components

The simulation results and the values of the structural metrics are plotted in diagrams to determine potential relations among them. Each dot presents one component. Figure 7-3 shows

four example diagrams. The two left diagrams indicate relations; the two right diagrams indicate no relations. In the upper left case two groups of components exist which differ both in the simulation results and in the values of the structural metrics. In the lower left case results and metrics form a “linear” relation. In the upper right case two groups of components differ in the simulation results but not in the metric values. In the fourth case no pattern is recognizable.

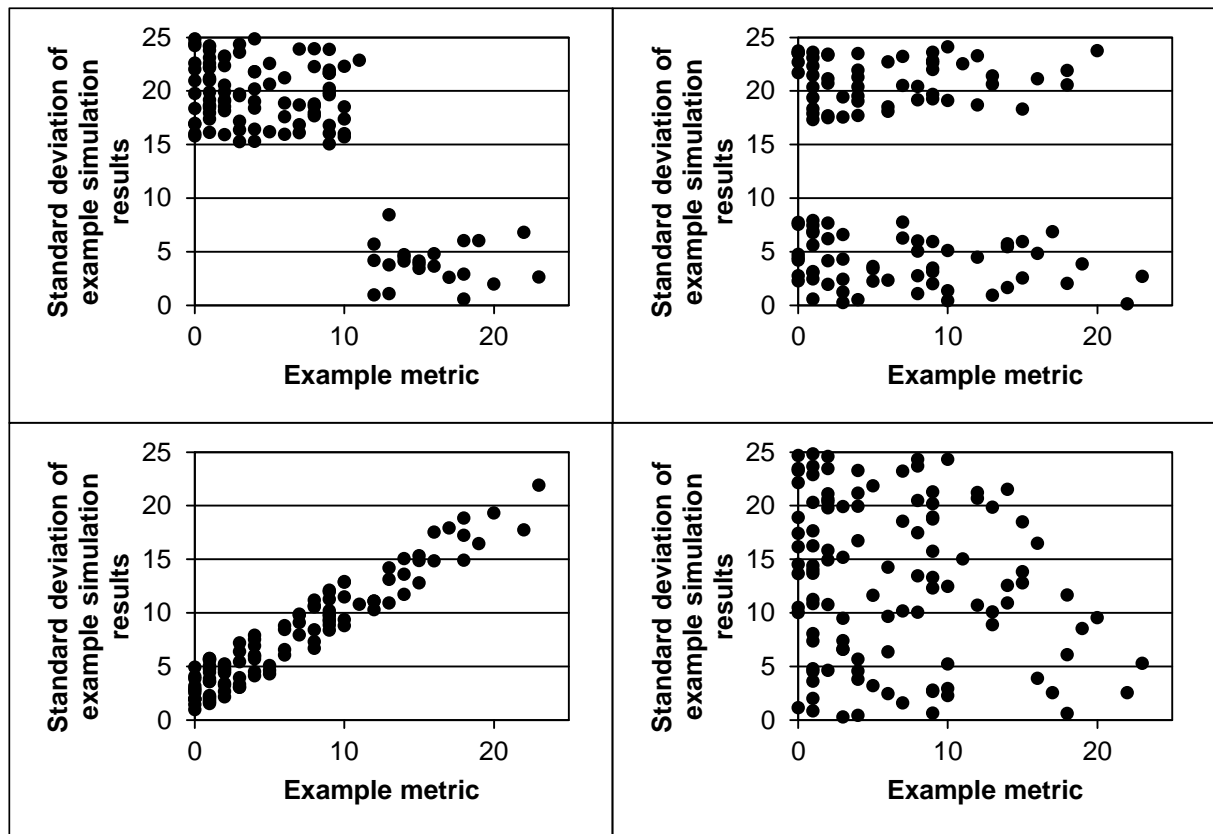


Figure 7-3 Example scatter plots for the relations between simulations results and structural metrics

The plots for the number of changed nodes and the number of time-steps hardly differ. This can be seen by row-wisely comparing the right and left column of Figure 7-4. Thus, all findings for the number of changed nodes apply for the number of time-steps and vice versa. This observation was not expected as the simulation allows for parallel execution of changes. It was assumed that the number of changed nodes and the number of time-steps are decoupled. However, the results show that they are highly correlated.

Components with high degree show small deviations (<5). This can be seen in the first row of Figure 7-4. Thus, components with high degrees show predictable, invariable behaviour. [BIEDERMANN ET AL. 2012] conjecture that highly connected nodes impose changes on their locality when changed, but are too constraint to have changes imposed on them when not. However, this observation is hardly generalizable as there are only two components with high degree (>12) in the case study at hand.

Components with high clustering coefficients show small deviations (values < 5). This can be seen in the second row of Figure 7-4. Thus, components with high clustering coefficients show predictable, invariable behaviour. [BIEDERMANN ET AL. 2012] conjecture that highly embedded nodes cannot impose changes on their locality when changed and are too constraint to have changes imposed on them when not. They assume that this observation is generalizable as there are 19 components with high clustering coefficient (values > 0.5) in the case study.

Components with high or low distance centrality show small deviations (values < 5). This can be seen in the third row of Figure 7-4. Thus, components with high or low distance centrality show predictable, invariable behaviour. [BIEDERMANN ET AL. 2012] have not conjectured what behaviour these components exhibit. However, this observation is hardly generalizable yet, as there are only one component with low distance centrality (values < 15) and only two components with high distance centrality (values > 40) in the case study at hand.

Components with low average distance show small deviations (values < 5). This can be seen in the fourth row of Figure 7-4. Thus, components with low average distance show predictable, invariable behaviour. However, this observation is hardly generalizable yet, as there is only one component with low average distance (values < 2) in the case study at hand.

There are two groups of components: one group with high deviations (values > 14 for number of changed nodes and values > 12 for number of time-steps) and one group with low deviations (values < 5 for both cases). This can be seen in all plots in Figure 7-4. Thus, there are two groups of components: one group with predictable, invariable behaviour and one group with unpredictable, variable behaviour. As there is quite a big gap between the groups, [BIEDERMANN ET AL. 2012] assume that there is a fundamental characteristic that differentiates the components of the groups. An efficient analysis, which allows for identifying the groups before modelling, simulating and analysing the behaviour, will greatly enhance the efficiency of system analysis. The low variety group should be modelled using simple models, which do not require a lot of data.

The groups do not correlate to one of the four metrics. This can be seen in any plot in Figure 7-4 as there is no clear separation of the groups on the x-coordinate. Thus, none of the metrics allows for separating the components into groups according to the predictability of the behaviour (i.e. the simulation results). Yet, [BIEDERMANN ET AL. 2012] conjecture that a separating criterion exists. It may be another structural metric or it may be based on the semantics of the models, e.g. the initial component size or a type of consistency relations.

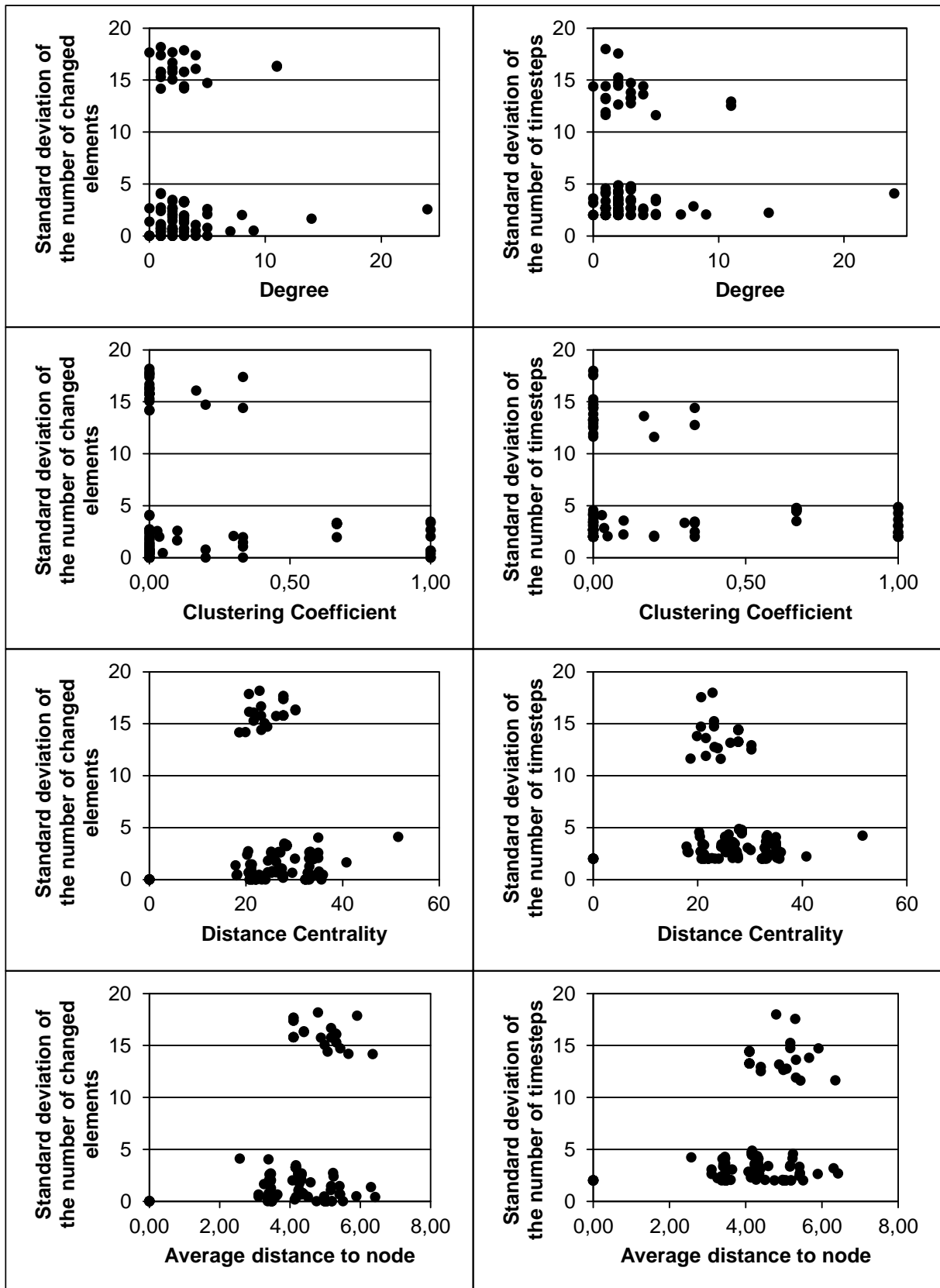


Figure 7-4 Scatter plots of the structural properties and change response variations

7.4 Power and limits of structural analysis for change prediction

Structural models are a suitable pre-work for building up simulation models. The results suggest that highly-connected (indicated by the degree) and highly-embedded (indicated by the clustering coefficient) components have very predictable dynamics. One explanation is that highly-connected components can impose changes onto their locality. Likewise highly-embedded components are too constraint to show highly variable behaviour. That means changing the component does not result in real modifications. Using degree and clustering coefficient for choosing appropriate dynamic models for the components is suggested. As high values indicate predictability these components should be modelled rather simple. Though the results show, that the behaviour is predictable, the actual behaviour has not been analysed. Thus, structural analysis allows for reducing the effort for creating simulation models, as some component, which required little data for modelling can be identified.

Moreover, the simulation results show that there are two groups of components: one group with high deviations and one group with low deviations. This indicates that there is an unknown variable determining the variability of the behaviour, which has to be identified in future research. Potential candidates for the unknown variable are the types or the mix of consistency relations, the (absolute or relative) size of the component or its neighbours. Other characteristics are not possible as they have not been added to the simulation model. Finding this variable would substantially improve change management, as it would allow for identifying components, which have to be handled and monitored carefully. Yet, structural analysis does not allow for identifying components with highly variable behaviour, i. e. components to be monitored and modelled carefully.

Due to the limited database the results are not generalizable. Yet, they allow for some insights on component change. Only the observations and conjectures concerning the clustering coefficient are generalizable. There are 19 components with high (values > 0.5) clustering coefficients. The other observations and conjectures are at best hints, which have to be validated and support by future research. Future research has to reassess the interaction between structural analysis, change simulation and real change management.

7.5 Validation of the minimal set of local metrics

The idea of the validation is to show that the minimal set provides the same insights as all available metrics. There are two levels of validation. The lower level simply means that the other metrics provide no new insights. The higher level means that the metrics of the minimal set provide the same insights as the corresponding cluster (see chapter 6 for the cluster definitions).

To validate the minimal set the insights of its metrics are compared to the insights of all metrics. Figure 7-5 shows the comparison for the metrics of the degree cluster including “distance centrality”. According to chapter 6 the “degree” metric should provide the same insights as the metrics: “path centrality”, “number of blocks per node”, “number of cliques per node”, “number of cycles per node” and possibly “distance centrality”.

As Figure 7-5 shows, the scatterplots for “path centrality”, “number of blocks per node” and “number of cliques per node” have the same basic structure as the one for “degree”: high metric values correlate with low behaviour deviations and low values do not correlate with either high or low behaviour deviations. Therefore, the three metrics provide the same insights as the degree metric. Hence, the minimal set is validated on a high level for these metrics.

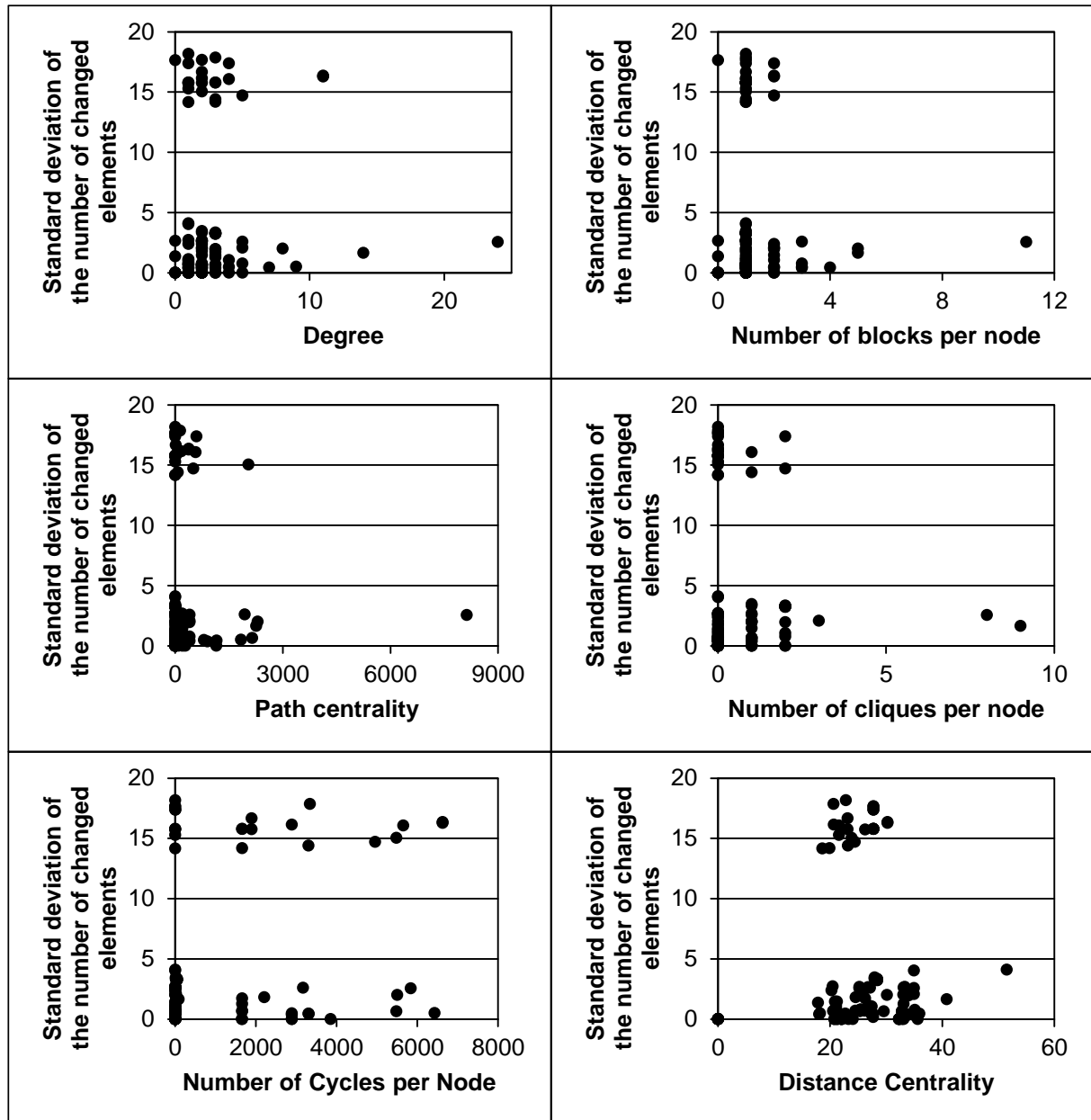


Figure 7-5 Comparison of the metrics in the degree cluster concerning change prediction

Interestingly, the number of cycles does not correlate with high (values > 14 for number of changed nodes and values > 12 for number of time-steps) or low (values < 5 for both cases) deviations. This can be seen in Figure 7-5. Thus, the number of cycles does not allow for predicting the variability of a component’s behaviour. This contrasts the general assumption

that many cycles indicate highly variable behaviour. Moreover, the metric provides less insight than the “degree”. Therefore, this case study only shows a lower level of validation for this metric replacement.

The final metric (“distance centrality”) provides other insights than the “degree” as discussed in the previous section. Thus, “distance centrality” should not be subsumed in the degree cluster and remain an independent metric in the minimal set.

7.6 Conclusion

This case study analyses, how structural characteristics impact change simulation results. The aim is to test if structural analyses can predict simulation results. If simulation results can be predicted the effort for creating simulation models can be reduced.

The study uses a novel simulation approach for change prediction based on cellular automata. The simulation runs cover the entire parameter space of the simulation model rather than using single scenarios or Monte-Carlo-approaches. Nevertheless the resulting data only allows for qualitative instead of statistical analysis.

The results show that behaviour (i. e. simulation results) can be deduced from structural models for some components. However, the behaviour of all components of a system cannot be predicted. Thus, dynamic models cannot be replaced by or derived from purely structural models. Based on the observations and considerations concerning generalizability, it is suggested to use the clustering coefficient for choosing appropriate dynamic models. High values indicate predictable behaviour, which can be described in simple models. However, the reduction of the modelling effort is rather low as the components with highly variable behaviour cannot be identified by structural analysis.

The validation analysis for the minimal set has two main results. First, the set is validated as the additional metrics provide no new and in most cases the same insights as the metrics from the minimal set. The only exception is the metric “number of cycles per node”, which provide fewer insights than its replacement (“degree”). This is unexpected as “cycles” in networks are usually associated with unpredictable behaviour. Second, the metric “distance centrality” provides other insights than the metric “degree”. Therefore, it should not be subsumed in the degree cluster and remain an independent metric in the minimal set.

8. Empirical analysis of concept reconfigurations

Chapter 6 proposes two minimal sets of structural metrics. This chapter validates the set for local metrics using an empirical approach. The idea is to show that the minimal set provides the same insights as all available metrics. There are two levels of validation. The lower level simply means that the other metrics provide no new insights. The higher level means that the metrics of the minimal set provide the same insights as the corresponding cluster (see chapter 6 for the cluster definitions).

The case study has to fulfil four main requirements according to subsection 5.3.5: application of structural analyses, relevant problem in engineering, proof of the significance of the structural analysis and proof of the validity of the minimal set.

Reconfiguration of production resources is a key measure to deal with changing products. Numerous models and metrics have been proposed to assess and support the reconfiguration of production resources. However, most of them have been derived from theory rather than from observation of real reconfigurations. This case study follows an empirical approach to define metrics for identifying production resources, which are likely to be changed during reconfigurations. According to [ZHANG ET AL. 2012]: “production [re]configuration entails a process of (1) identifying the relevant process elements, such as routings, operations and manufacturing resources (e.g., machines, tools, fixtures) for the given product variants, (2) configuring production processes from the identified process elements, and (3) selecting appropriate processes based on the evaluation of the multiple alternatives configured.” Production resources can be designed for reconfiguration at higher costs during development and production of the resources. Therefore, the need for reconfiguration needs to be assessed during the concept phase of the resource life-cycle. Thereby, a suitable design strategy to incorporate reconfigurability into the production resource can be defined. Figure 8-1 shows the interrelation among the resource life-cycle, design for reconfiguration and the role of structural analysis.

The case study uses an empirical approach (see subsection 4.2.4) to show the significance of the results. The basic hypothesis is: highly interlinked components are often changed. To test the hypothesis reconfigurations of two products are analysed retrospectively. The production resources and reconfigurations are modelled as a network of interlinked components. Then the structural properties of changed, removed, extended and unchanged components are compared to check if they significantly differ. Thereby, metrics are identified, which allow for assessing the need for configuration.

This case study has already been partially published in [BIEDERMANN ET AL. 2010]. The structural modelling of the products and changes are also described in the theses [STRELKOW 2010], [GÜNTHER 2011], [KONG 2011] and [MANTEUFFEL 2011], which are not publicly available. Moreover, the study uses preliminary work by [ZAEH ET AL. 2010] and [ZAEH ET AL. 2011]. To keep the presentation brief and focused many details of the structural models and the reconfiguration models are omitted in the text but are provided in section 11.8 of the appendix. Moreover, the publications above are referred to if appropriate or necessary.

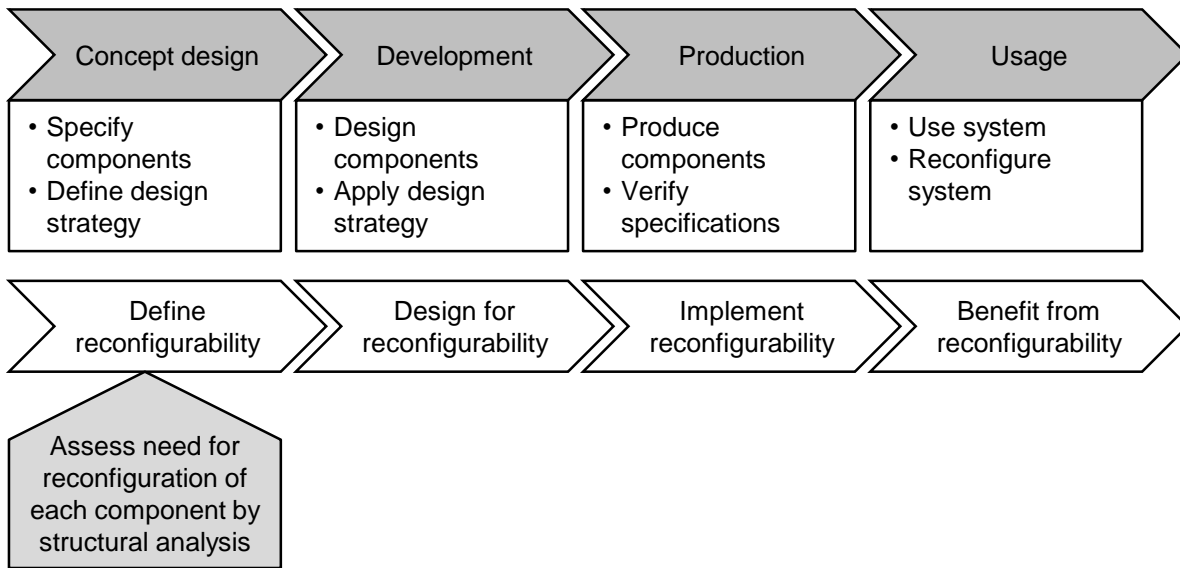


Figure 8-1 Design and reconfiguration in the life-cycle of a production resource

8.1 Modelling of concept reconfigurations

Due to varying product portfolios and optimisations of resource consumption production systems are routinely reconfigured in their life-cycle today. To allow for efficient reconfigurations production systems are designed to incorporate flexibility in their layout, their components and their production programs. One of the key challenges when designing the production system is to identify those components, which are most likely to be changed and require flexible designs. This study examines if structural metrics of the component network allow for identifying these components by analysing past reconfigurations. This case study has two research questions.

- Do changed components or removed components differ structurally from unchanged components?
- Do extended (i.e. a new component has been added to them as a neighbour) components differ structurally from not extended components?

There are four modes of component modifications considered in this study: add, change, replace and remove. A component is added if new functionalities are required after the reconfiguration. A component is changed if its functionality needs to be slightly modified. A component is replaced if its functionality needs to be substantially modified (and replacing it is more economical). A component is removed if its functionality is no longer required. These modes are reflected in modifications of the component structure as shown in Figure 8-2. However, the modes replace and change cannot be structurally differentiated. Therefore, they are treated as one mode in this study.

The method for modelling the reconfigurations has been developed as a tool for planning the reconfigurations (see [ZAEH ET AL. 2010]). Basis is a network model of the entire production

resource. The reconfigurations are iteratively modelled starting with the components, which are directly affected by the cause for the reconfigurations. The network model serves then as guiding tool for determining those components, which are affected by knock-on changes. The model of the configuration is a substructure of the resource network. Though, this modelling approach was originally intended to serve as part of a planning tool (including the assessment with key performance indicators [ZAEH ET AL. 2011]) it can also model past reconfigurations.

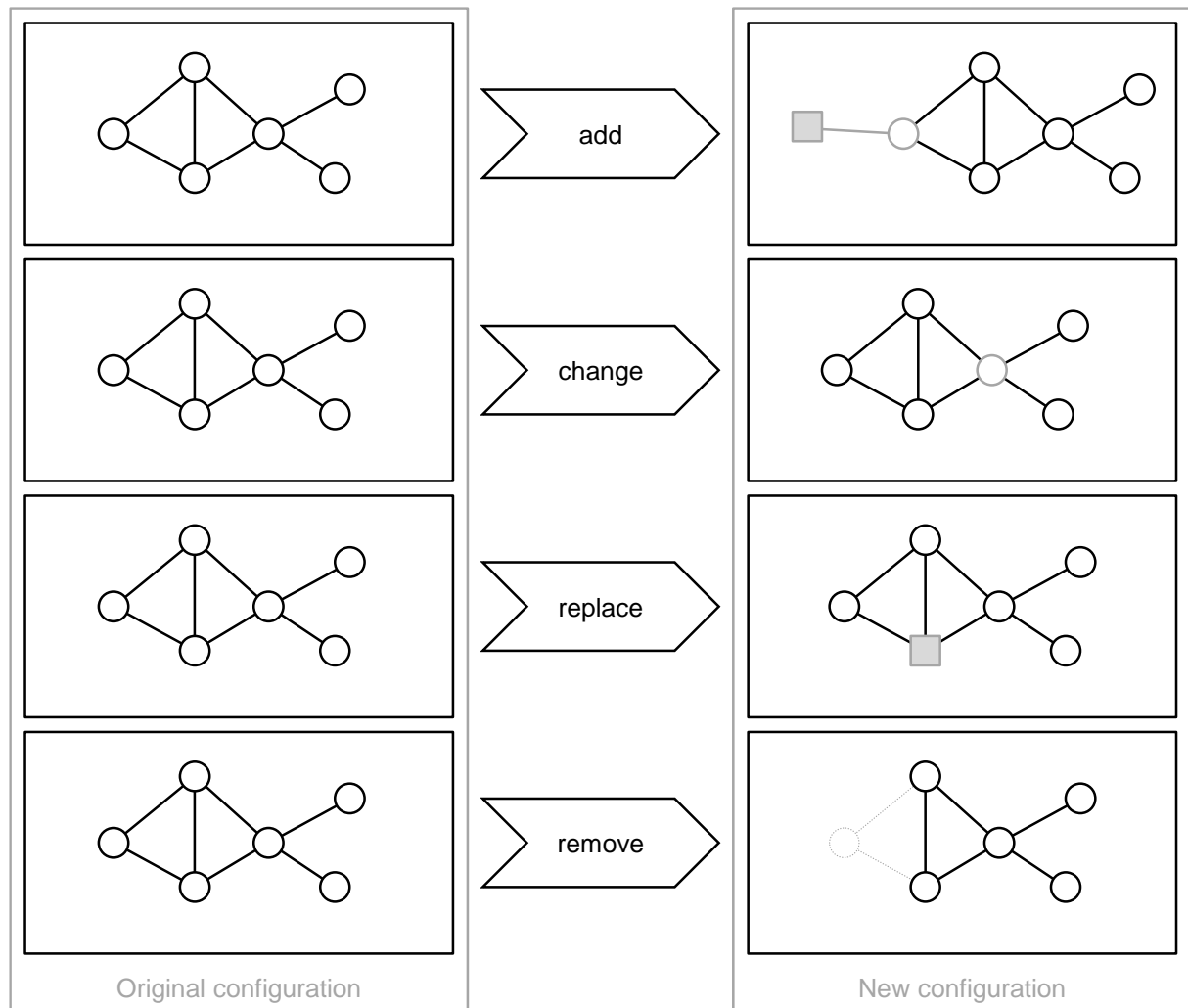


Figure 8-2 Types of production cell reconfigurations

8.2 Observation of production cell reconfigurations

The previous section defined the scope of this study and stated the research question. This section describes the research methodology. The basic research idea of this study comprises four steps: modelling several products using component networks, modelling several reconfigurations (see the previous section for details), comparing the structural characteristics of the affected components and testing the significance of the observations statistically.

Two products are analysed in this study: an assembly cell and a manufacturing cell. The assembly cell is part of a research facility for smart production systems. It has been reconfigured once to fit new research aims and test new technologies. The manufacturing cell is part of a continuous production process in the semi-conductor industry. It has been reconfigured twice to optimise the production process. Both cells have been modelled using product structures to analyse their reconfigurations. The assembly cell model comprises two levels of granularity (module level and component level) to allow for comparing the results for these levels. The manufacturing cell has only been modelled at module level due to time limitations. All models comprise several types of relations (contact, cable, signal, electrical, pneumatics and material). However, only the contact and cable relations are analysed in this study as they comprise all other relation in this case. The details of the modelling processes are described in the theses [STRELKOW 2010], [GÜNTHER 2011], [KONG 2011] and [MANTEUFFEL 2011].

The reconfiguration of the assembly cell modified it to allow for producing a new type of product. The first reconfiguration of the manufacturing cell incorporated a new handling strategy. The second reconfiguration of the manufacturing cell increased the product size by 50%. For each reconfiguration the modifications of the product structures have proposed determined according to the method by [ZAEH ET AL. 2010] (see also the previous section). The added, the changed and the removed components are documented as well as the unchanged components. The numbers for each combination of reconfiguration and level of abstraction are shown in Table 8-1 (please note: if a new component was added to a module, the module is considered to have changed). Additionally, the structures before and after the reconfigurations were compared to determine, which components had been extended (i.e. have been connected to a newly added component). The details of the reconfigurations are described in the theses [GÜNTHER 2011], [KONG 2011] and [MANTEUFFEL 2011].

Table 8-1 Number of production cell reconfigurations

No.	Reconfiguration	Level of abstraction	Add	Change	Remove	None
A	Product change	Component	17	14	6	71
B	Product change	Module	4	6	3	8
C	New handling strategy	Module	0	5	0	36
D	Scale-up of product	Module	0	20	0	21

The components of the cells were structural characterised by the four metrics of the minimal set (see chapter 6): degree, clustering coefficient, distance centrality and average distance to node. Additionally, six metrics were computed to allow for testing the validity of the minimal set: blocks per node, cliques per node, cycles per node, maximum distance to node, median distance to node and path centrality. The metrics were computed with LOOMEOTM 2.5.0 and Microsoft ExcelTM 2010 (see also Table 5-1). To compare the characteristics of changed, removed and unchanged components the average values for each group, each metric and each

reconfiguration were computed. The comparison between extended and not extended component was computed analogue. The comparisons of the average values are shown in Figure 8-3 and Figure 8-4 for reconfiguration B. Figure 8-5 and Figure 8-6 show the comparison between the minimal set and the remaining metrics. The remaining comparisons are shown in section 11.8 of the appendix.

Finally, the observed differences between the average metric values are tested for significance by the Welch-Test [WELCH 1947]. The Welch-Test is used and the variances of the metrics are not homogeneous and the metrics are assumed to have a normal distribution. The statistical computations were done with Microsoft Excel™ 2010.

8.3 Structural properties of reconfigured components

This section presents the comparison of the structural characteristics between changed, removed and unchanged components. Figure 8-3 shows the comparison for the reconfiguration of the assemble cell at module level. The comparison for the over three reconfigurations are shown in section 11.8 of the appendix. Moreover, Figure 8-3 shows only the metrics of the minimal set for local metrics. The comparison to all metrics is shown in Figure 8-5 and discussed in section 8.5. The analysis of the charts leads to following observations:

1. Changed components have the highest average degree.
2. Removed components have the lowest average degree.
3. Unchanged components have highest average clustering coefficient.
4. Changed and removed components have the same average clustering coefficient.
5. Changed components have the lowest average of the average distance.
6. Removed components have the highest average of the average distance.
7. Unchanged components have lowest average distance centrality.
8. Changed and removed components have the same average distance centrality.

The observations 1, 3, 5 and 7 are confirmed in the other three reconfigurations. The observations 2, 4, 6 and 8 are only confirmed for the reconfiguration of the assemble cell at component level as the reconfigurations of the manufacturing cell did not involve the removal of module. None of the observations is statistically significant for all reconfigurations. Yet, most of them are significant in at least on reconfiguration. Therefore, they and the conclusions drawn from them must be treated with some caution. The detailed results of the significance tests are documented in section 11.8 of the appendix.

The charts also show that degree and average distance provide the same insights though the charts are mirrored to a certain degree. The same holds for clustering coefficient and distance centrality. The analysis of all charts also shows that the differences for changed, removed and unchanged components are more pronounced for the metrics degree and clustering coefficient. The comparison of the charts for the assembly cell reconfiguration at module level and at component level shows that the observations are qualitatively the same, yet, more pronounced at component level.

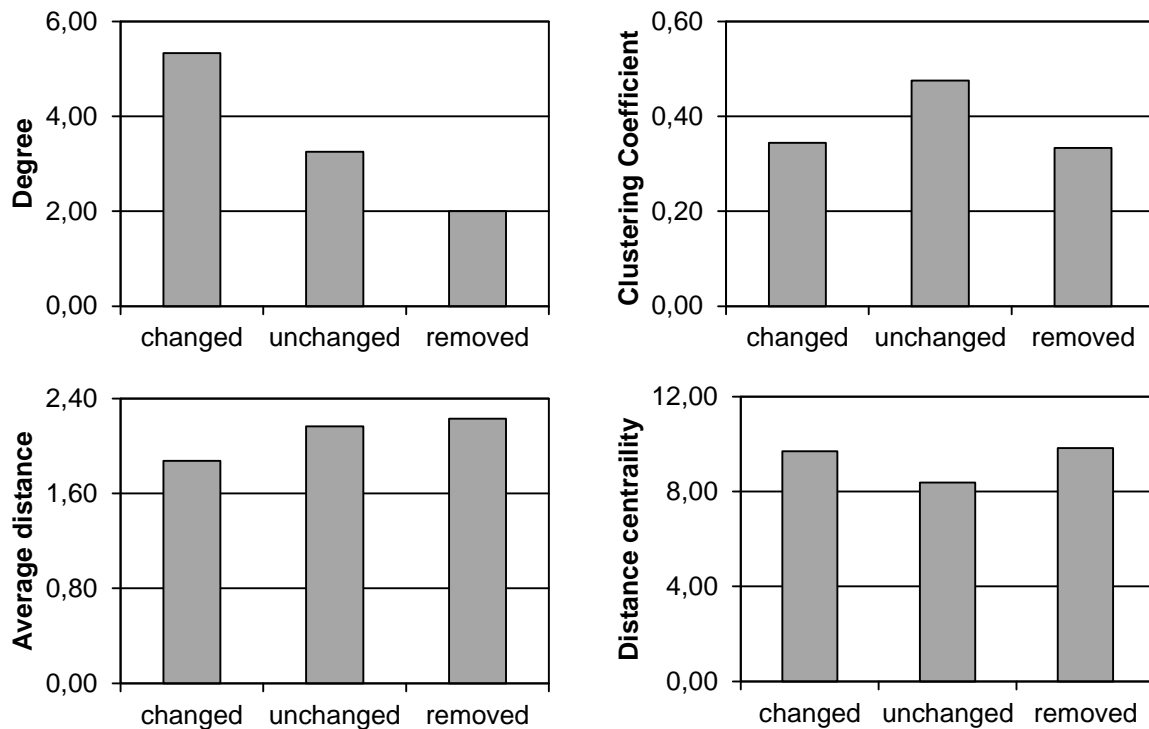


Figure 8-3 Comparison of changed, removed and unchanged components for the product change at module level

Figure 8-4 shows the comparison for the reconfiguration of the assemble cell at module level. The comparison for the over three reconfigurations are shown in section 11.8 of the appendix. Moreover, Figure 8-4 shows only the metrics of the minimal set for local metrics. The comparison to all metrics is shown in Figure 8-6 and discussed in section 8.5. The analysis of the charts leads to following observations:

1. Extended components have a higher average degree.
2. Extended components have a lower average clustering coefficient.
3. Extended components have a lower average of the average degree.
4. Extended components have a higher distance centrality.

The observations are confirmed for the reconfiguration of the assembly cell at component level. The reconfigurations of the manufacturing cell do not add modules. Therefore, the observations can be neither confirmed nor rejected.

The comparison of the charts for the assembly cell reconfiguration at module level and at component level shows that the observations are qualitatively the same, yet, more pronounced at component level.

The charts also show that all metrics provide the same insights. This is expected as only two classes of components need to be differentiated. The metrics differ only the relation between component class and value range. Extended components have higher value of the degree and distance centrality and lower values of the clustering coefficient and average distance. Therefore, one metric already allows for identifying potentially extended components. Two

criteria are relevant for choosing the appropriate metrics: the relative difference of the value and the statistical significance of the differences. Thus, degree and clustering coefficient are the best candidates. They can also be combined to create more pronounce classifications.

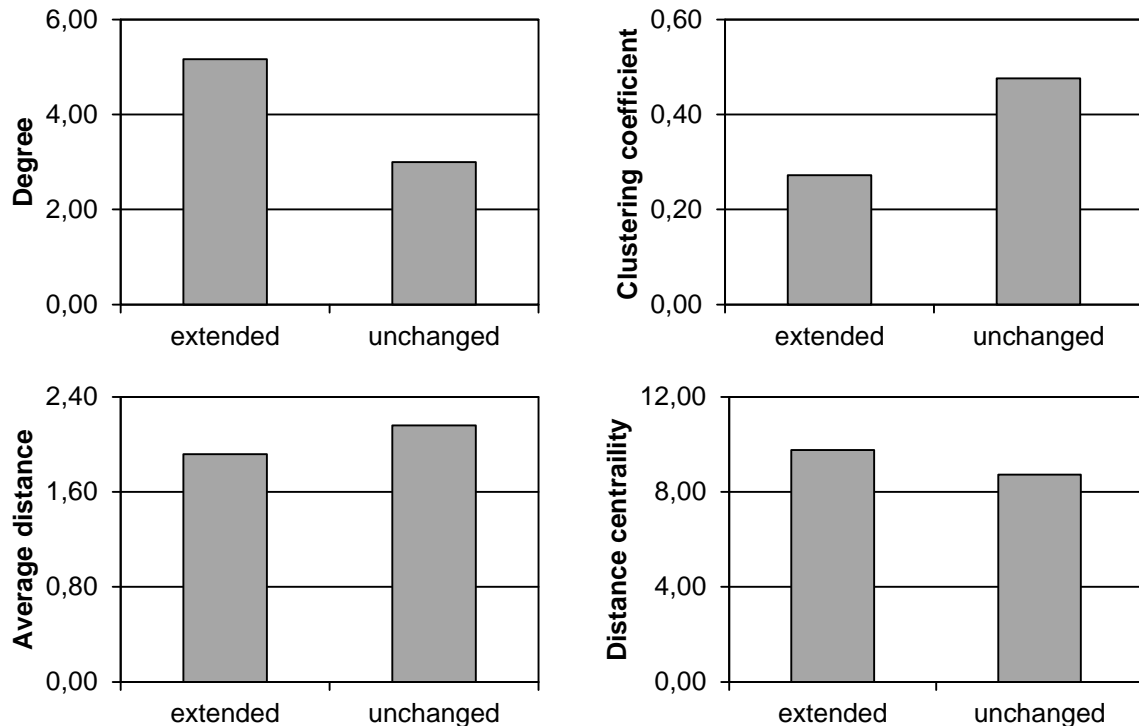


Figure 8-4 Comparison of extended and unchanged components for the product change at module level

8.4 Prediction of reconfigurations with structural metrics

The main result of this study is: structural analysis allows for identifying components, which are likely to be changed, removed or extended during reconfigurations of production resources due to optimisations. Components, which are likely to be changed or to be extended, have high values of the degree and rather low values of the clustering coefficient. Components, which are likely to be removed, have low values of the degree and rather low values of the clustering coefficient. Components, which are likely to remain unchanged, have rather median values of the degree and rather high values of the clustering coefficient.

The classification of the components allows for choosing an appropriate design strategy. Components may be designed for changeability if they are commonly changed. Components may incorporate standardised interfaces to allow for extensions. Other components may be standardised as they are unlikely to be affected by reconfigurations.

However, the findings cannot be directly used to create tool to classify components of production resources. The results provide no general rule what “high”, “median” or “low” values of the metrics are. The clustering coefficient is a normalised metric. Therefore, 0.5 is a natural candidate for separating “high” and “low” values as it is the middle between the

theoretical maximum (1.0) and minimum (0.0). This adequacy of the value still has to be shown in future research. The metric degree is more problematic. First, three value ranges (“high”, “median” or “low”) need to be separated. Second, the metric is not normalised and may depend on the total number of components. The separation of the value ranges can be linked to the average degree. Yet, sensible value ranges need to be identified in future research. Another important open point is the appropriate visualisation of the metrics. As two metrics are proposed, a portfolio diagram is an obvious proposal.

Future research also has to address the generalizability of the results. The study has only addressed production resources, which have been reconfigured due to optimisation requests. The results may be generalised in three aspects: modification reason, type of modification and type of system. The classification of the changed, removed and extended components may also apply to other reasons such as legal or localisation requests. It may also apply to general changes not just reconfigurations. It may also apply to general hardware products not just production resources, which generalisations are valid is a matter of conjecture.

8.5 Validation of the minimal set

The idea of the validation is to show that the minimal set provides the same insights as all available metrics. There are two levels of validation. The lower level simply means that the other metrics provide no new insights. The higher level means that the metrics of the minimal set provide the same insights as the corresponding cluster (see chapter 6 for the cluster definitions).

To validate the minimal set the insights of its metrics are compared to the insights of all metrics. Figure 8-5 and Figure 8-6 show the comparisons for the metrics of the degree cluster including “distance centrality” for the reconfiguration of the assembly cell at module level. The comparisons for the other three reconfigurations are shown in section 11.8 of the appendix. According to chapter 6 the “degree” metric should provide the same insights as the metrics: “path centrality”, “number of blocks per node”, “number of cliques per node”, “number of cycles per node” and possibly “distance centrality”.

As Figure 8-5 shows, the charts for “path centrality”, “number of cliques per node” and “number of cycles per node” have the same basic structure as the one for “degree”: changed components have high values, unchanged components have median values and removed components have low values. Therefore, the three metrics provide the same insights as the degree metric. Hence, the minimal set is validated on a high level for these metrics.

The chart for the metric “number of blocks per node” differs from the degree chart. Changed components still have the highest average value, but unchanged and removed components have the same average value (1.0). However, this result does not contradict the validity of the minimal set as 1.0 is the lowest theoretically possible value for the “number of blocks per node”. Therefore, the minimal set is still valid. The final metric (“distance centrality”) provides other insights than the “degree” as discussed in the previous section. Thus, “distance centrality” should not be subsumed in the degree cluster and remain an independent metric in the minimal set. The comparisons among the metrics of the degree cluster for the other three reconfigurations confirm the minimal set as well.

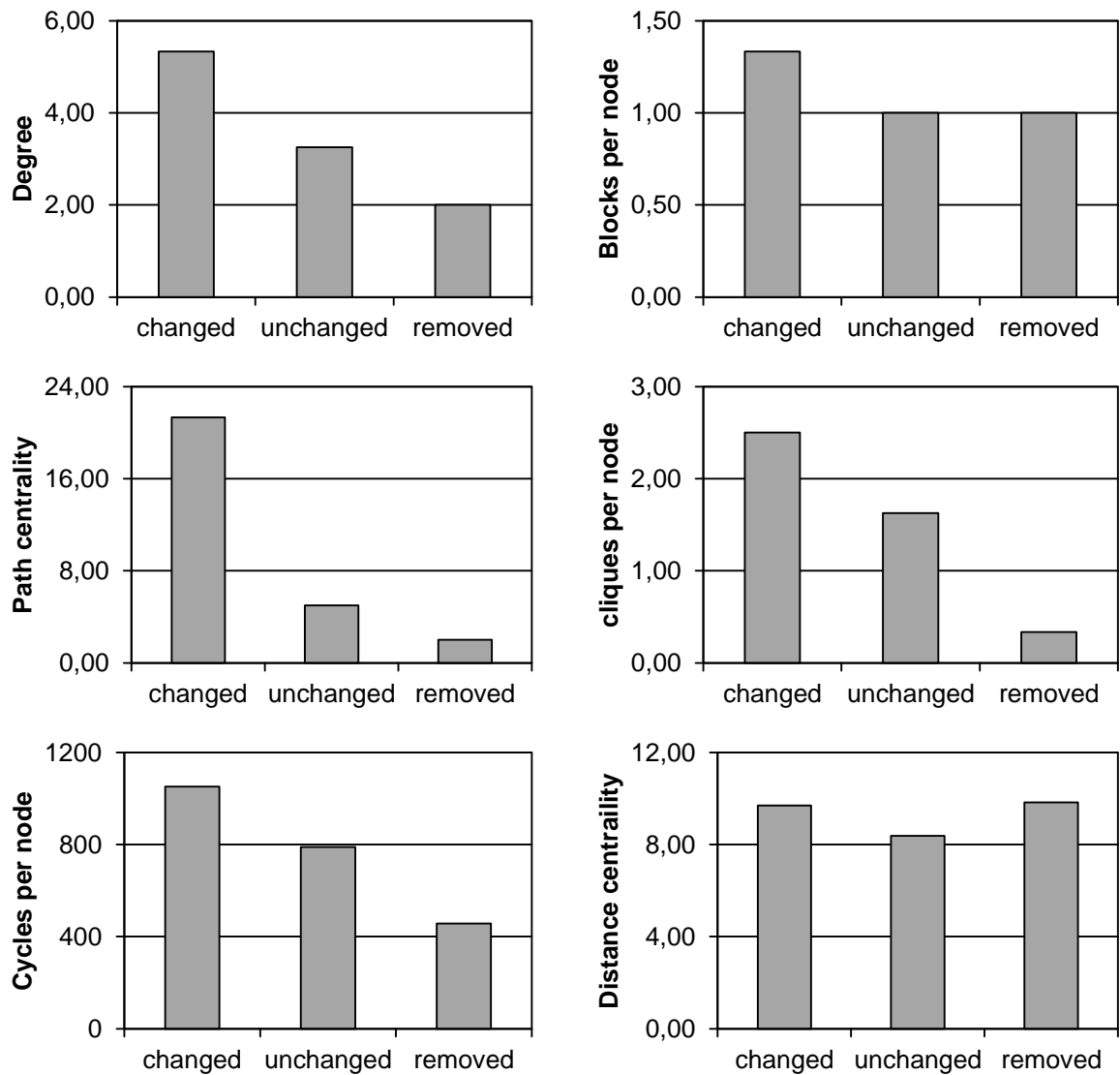


Figure 8-5 Comparison of the metrics in the degree cluster for changed, unchanged and removed components for the product change at module level

Figure 8-6 shows the comparison among the metrics of the degree cluster for extended and not extended components. All charts have the same structure: extended components have higher average metric values. Yet, the differences vary in their clarity. In particular, the metrics “number of blocks per node” and “distance centrality” show much less pronounced differences. Nevertheless, the minimal set is validated on a high level for all metrics of the degree cluster. The comparisons among the metrics of the degree cluster for the other three reconfigurations confirm the minimal set as well.

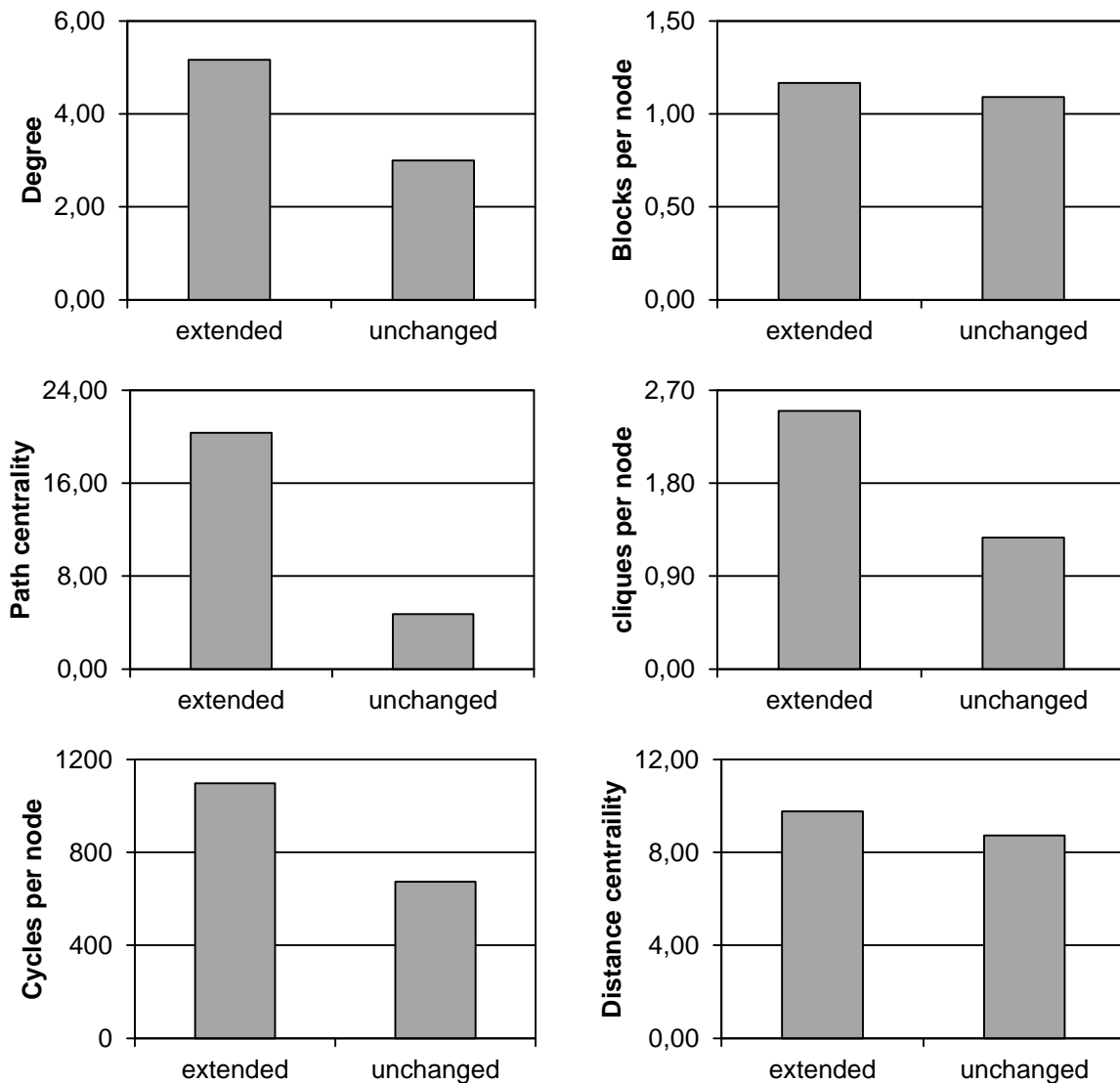


Figure 8-6 Comparison of the metrics in the degree cluster for extended and unchanged components for the product change at module level

8.6 Conclusion

The validation analysis for the minimal set has two main results. First, the set is validated as the additional metrics provide no new and in most cases the same insights as the metrics from the minimal set. Second, the metric “distance centrality” provides other insights than the metric “degree” for the changed and removed components but the same insights for extended components. Therefore, it should not be subsumed in the degree cluster and remain an independent metric in the minimal set.

The results show that the structural characteristics of a component and the likelihood to be changed, removed or extended are highly correlated. Components, which are likely to be

changed or to be extended, have high values of the degree and rather low values of the clustering coefficient. Components, which are likely to be removed, have low values of the degree and rather low values of the clustering coefficient. Components, which are likely to remain unchanged, have rather median values of the degree and rather high values of the clustering coefficient. The structural metrics allow for classifying the components according to the likely future modifications and Thus, for choosing an appropriate design strategy.

The findings cannot be directly used to create tool to classify components of production resources. The results provide no general rule what “high”, “median” or “low” values of the metrics are. Another important open point is the appropriate visualisation of the metrics. Future research also has to address the generalizability of the results. The study has only addressed production resources, which have been reconfigured due to optimisation requests. The results may be generalised in three aspects: modification reason, type of modification and type of system, which generalisations are valid, is so far a matter of conjecture.

9. Conclusion and Outlook

The overall aim of this thesis is to provide more guidance in structural analysis as this is often requested from industry. From an industrial perspective structural analysis is too complicated and too time-consuming although the achieved insights and results are acknowledged. Both, complicatedness and time consumption result from the flexibility of the modelling approach and the number of available analyses. From a scientific point of view research on structural analysis has been mostly exploratory so far and has produced mostly qualitative results. Therefore, this thesis aims at consolidating parts of the research on structural analysis.

The consolidation process results in two metrics of the minimal set sets for the analysis of product components networks. One set allows for analysing individual components within the network; the other one allows for analysing entire networks. The sets are the result of rigorous tests of the applicability criteria using empirical and statistical methods. Initial validation in two case studies indicates the validity of the sets. By reducing the number of sensible analyses the results make structural analysis and modelling simpler. Moreover, the thesis introduces structural analysis scenarios – a documentation scheme for structural analyses, which provides a basis for analysis guidelines.

Yet, the state of the art shows a lack of empirical validation of both the predictions and the feasibility of structural analysis. The main reason is the time-consuming modelling processing for most structural models. The time and the effort needed for collecting data and creating structural models limit the models available to research. Therefore, mostly single case studies are provided and hardly any comparison of results is possible. Improving the modelling methods in terms of accuracy and time consumption is a pressing issue, which is not addressed in this thesis. More efficient modelling would allow for more rigorous research by providing data for empirical and statistical research methods.

This would allow for resolving one of the most pressing issues of structural analysis research: a comparison to other analysis approaches in terms of reliability, effort and availability. So far, it is not possible to tell if structural analysis is the best option in any situation. Other approaches e.g. system dynamics or even Delphi analyses might be more useful. Which option is the best, is at the moment a matter of conjecture or individual convictions.

Finally, the fundamentals sets have so far only been identified and tested for hardware component networks. The main assumption – the existence of a minimal set – should hold true for other types of systems as well but still needs to be researched. Among the types to be tested are software component networks, organisation networks, process networks and parameter networks.

9.1 Metric of the minimal set sets for product concepts

The thesis presents an approach to test the applicability of structural analyses and to identify metric of the minimal set sets. All structural analyses have to fulfil three main criteria to be applicable: computability, disparity and significance. Computability means that an analysis can

be performed on a given model. Disparity means that analyses must have a suitable value ranges and differ significantly from each other. Significance means that the analysis allows for solving the initial problem. The criteria differ in terms of the corresponding testing methods. Whereas computability can be tested with a checklist, disparity tests require meta-analyses of a range of models and significance can be tested with various methods e.g. analogy or simulation.

The disparity test is a way to determine candidates for metric of the minimal set sets. It requires the analysis of several models of one type to determine correlations among the metric values. By eliminating highly correlated metrics candidates for minimal sets are derived. The approach is applied to concept models of component networks and results in two sets: one for local metrics and one for global metrics. In total ten local and sixteen global metrics are tested for disparity. This thesis uses two model collections for the meta-analysis. The first collection is derived from literature and comprises 35 models. The second collection is derived from a design repository and comprises 124 models. Both collections cover about the same range models sizes but differ slightly in level of abstraction and type of product.

The global metric set is slightly ambiguous. Two metrics (“average number of blocks per node” and “number of cliques”) may not be required in the minimal set. However, the analyses of the two collections contradict each other for the two metrics. Therefore, both should be tested in future research. Thus, the minimal set of global metrics has to be carefully assessed. So far, it comprises six metrics:

- average degree
- average number of blocks per node
- number of cliques
- number of cycles
- number of edges
- number of nodes

The local metric set is less ambiguous. However, the metric “distance centrality” should be tested if it can be replaced by the metric “degree”. So far, the minimal set of local metrics comprises the four metrics:

- degree
- clustering coefficient
- distance centrality
- average distance to node

By identifying the minimal sets the requirement for disparity among the metrics is proved for component network models of hardware products. The minimal sets need to be tested in real applications to show that they really provide the same analytical power as all metrics. The thesis provides two case studies for the initial validation of the minimal sets.

The identification of the minimal sets simplifies structural analysis as fewer metrics need to be understood, researched and applied. However, it does not improve the guidance when applying structural metrics. Future research has to document the significance of the metrics in various applications of structural analysis. In particular, the assessment of life-cycle properties (e.g. changeability) and product-inherent properties (e.g. modularity) needs to be consolidated. This

would result in guidelines for applying structural metrics, which in turn would increase the applicability in industrial contexts.

9.2 Empirical validation of structural concept analysis

This section focuses on the validation via testing the significance of the minimal sets. The minimal sets must provide the same analytical power as all metrics together. Otherwise the sets are not valid and must be changed, extended or dismissed. This thesis provides two case studies to test the validity of the minimal set. The first case study analyses how structural characteristics impact the change simulation results. The study uses a simulation approach to show the significance of the results. The second case study addresses the assessment of the reconfigurability of production resources. The study uses an empirical approach to show the significance of the results.

The results of the first case study show that behaviour (i. e. simulation results) can be deduced from structural models for some components. However, the behaviour of all components of a system cannot be predicted. Thus, dynamic models cannot be replaced by or derived from purely structural models. The results of the second study show that the structural characteristics of a component and the likelihood to be changed, removed or extended are highly correlated. The structural metrics allow for classifying the components according to the likely future modifications and thus, for choosing an appropriate design strategy.

However, the findings cannot be directly used to create a tool to classify hardware components. The results provide no general (i.e. size-independent) value ranges. Another important open point is the appropriate visualisation of the metrics. Future research also has to address the generalizability of the results. The studies have only addressed production resources. Which generalisations are valid is so far a matter of conjecture.

The validation analysis for the minimal set has two main results. First, the set is validated as the additional metrics provide no new and in most cases the same insights as the metrics from the minimal set. Second, the metric “distance centrality” sometimes provides other insights than the metric “degree”. It should not be subsumed in the degree cluster and remain an independent metric in the minimal set. Therefore, the initial validation shows that the minimal set for local metrics is valid. The set for global metrics needs to be validated in future research.

Moreover, the two case studies show the effort needed for achieving significant research results. The results of the first study are debatable as only one system was simulated. Yet, a lot of modelling effort for creating the structural and the simulation model as well as simulation time went into it. The results of the second study are partially statistically significant. It required a lot of effort for creating the structural models and documenting the reconfigurations. Therefore, the modelling methods need to be improved in terms of accuracy and time consumption. More efficient modelling would allow for more rigorous research by providing data for empirical and statistical research methods.

9.3 Structural analysis of non-product systems

This thesis focuses on structural modelling and analysis of hardware components due to its research background. A literature review reveals, that most published models base on contact relations even though a wide variety of other relations has been introduced. Thus, this thesis identifies metrics of the minimal set for component networks describing contact relations. The discussion of the interactions between several types of relations shows that contact relations are sometimes a prerequisite for other types of relations show as energy flows. Therefore, the findings should also apply to other component networks e.g. modelling functional relations. The results show that the main hypothesis of this thesis – existence of metrics of the minimal set sets – holds true for components networks. It is assumed that the same sets also apply to general product structures based on hardware components.

Due to the flexibility of structural modelling several other types of structures have been analysed – software structures, organisational structures, process structures and parameter structures. The research methodology of this thesis can be easily transferred to other types of structure. Its main steps are collect a representative set of structural models, apply all metrics to them, analyse the correlations among the metric values, derive candidates of minimal sets and validate the sets in industrial applications. This leads to the final conjecture of this thesis: metric of the minimal set sets exist for software structures, organisational structures, process structures and parameter structures as well.

If these structures possess the same minimal sets as hardware structures is a matter of conjecture. Organisational structures might have the same minimal sets as they are commonly modelled as undirected communication networks. Software structures, process structure and parameter structures should have different sets as they are usually modelled as directed networks. If all types of structures have the same minimal sets this is a valuable insight for network theory as it assumes that all networks follow the same laws of nature.

9.4 Comparison of structural analyses to other approaches

One particular drawback of research on structural analysis is not resolved in this thesis: the lack of comparison to other approaches. Structural analysis has been applied in a wide range of applications across the product lifecycle: project planning, concept design and evaluation, process management, variant management and many others. For most application, other methods exist – e.g. system dynamics or methods from operations research. However, so far hardly any comparison among the approaches has been published.

The comparison should evaluate the approaches in terms of reliability, effort and availability. Reliability comprises the accuracy of the results, the reproducibility of the analyses and the usefulness of the models. Effort comprises all resources, which are necessary to perform the analysis including training costs. Availability comprises the point of time at, which the required data becomes available in the product life cycle. So far, it is not possible to tell if structural analysis is the best option in any situation. Other approaches e.g. system dynamics or even Delphi analyses might be more useful, which option is the best, is at the moment a matter of conjecture and/or individual convictions.

10. References

[AKAIKINE 2010]

Akaikine, A.: The Impact of Software Design Structure on Product Maintenance Costs and Measurement of Economic Benefits of Product Redesign. Master Thesis, Massachusetts Institute of Technology, 2010.

[ALIZON ET AL. 2006]

Alizon, F.; Shooter, S.B.; Simpson, T.W.: Improving an Existing Product Family Based on Commonality/Diversity, Modularity and Cost. In: Proceedings of IDETC/CIE 2006 ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 10-13, 2006, Philadelphia, Pennsylvania, USA, DETC2006-99536.

[AMERI ET AL. 2009]

Ameri, F.; Summers, J.D.; Mocko, G.M.; Porter, M.: Engineering Design Complexity: An Investigation of Methods and Measures. In: Research in Engineering Design, Vol. 19, 2008, pp. 161-179. DOI: 10.1007/s00163-008-0053-2

[ARIYO ET AL. 2007]

Ariyo, O.O.; Keller, R.; Eckert, C.M.; Clarkson, P.J.: Predicting Change Propagation on Different Levels of Granularity: an Algorithmic View. In: Bocquet, J.-C. (ed.): Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007, No. 220. ISBN 1-904670-02-4.

[ARTS ET AL. 2008]

Arts, L.; Chmarra, M.K.; Tomiyama, T.: Modularization Method for Adaptable Products. In: Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2008 August 3-6, 2008, Brooklyn, New York, USA, DETC2008-49338.

[BADOR ET AL. 2007]

Bador, D.P.M.D.; Seering, W.J.; Rebentisch, E.S.: Measuring the Efficiency of Commonality Implementation: Applications to Commercial Aircraft Cockpits. In: Bocquet, J.-C. (ed.): Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007, No. 395. ISBN 1-904670-02-4.

[BAGLEY 2005]

Bagley, M.R.: Modeling an Automobile Steering System Using Axiomatic Design's Design Matrix and the Design Structure Matrix. Master Thesis Massachusetts Institute of Technology, 2005.

[BALDWIN & CLARK 2000]

Baldwin, C.Y.; Clark, K.B.: Design Rules: The Power of Modularity. MIT Press 2000. ISBN: 978-0262024662.

[BARTOLOMEI 2007]

Bartolomei, J.E.: Qualitative Knowledge Construction for Engineering Systems: Extending the Design Structure Matrix Methodology in Scope and Procedure. PhD Thesis Massachusetts Institute of Technology, 2007.

[BARTOLOMEI ET AL. 2011]

Bartolomei, J.E.; Hastings, D.E.; de Neufville, R.; Rhodes, D.H.: Engineering Systems Multiple-Domain Matrix: An Organizing Framework for Modeling Large-Scale Complex Systems. In: Systems Engineering, Vol. 15, No. 1, 2012, pp. 41-61.

[BATTISTA ET AL. 1998]

di Battista, G.; Eades, P.; Tamassia, R.; Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentice Hall 1998. ISBN: 978-0133016154

[BAUER & MAURER 2011]

Bauer, W.; Maurer, M.: Future-proof Interfaces: Systematic Identification and Analysis. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USE, 14-15 September 2011, pp. 97-109. ISBN 978-3446-43037-2.

[BAUER ET AL. 2011]

Bauer, W.; Daniliidis, C.; Lindemann, U.: Approach for a Modularization Driven System Definition Using Multiple Domains. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USE, 14-15 September 2011, pp. 85-96. ISBN 978-3446-43037-2.

[BIEDERMANN & LINDEMANN 2011A]

Biedermann, W.; Lindemann, U.: Designing Consistent Structural Analysis Scenarios. In: Culley, S. J.; Hicks, B. J.; McAlloone, T. C.; Howard, T. J.; Lindemann, U. (Eds.): Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 4: Product and Systems Design, Lyngby/Copenhagen, Denmark, 15.-19.08.2011, pp. 133-144. ISBN: 978-1-904670-24-7.

[BIEDERMANN & LINDEMANN 2011B]

Biedermann, W.; Lindemann, U.: On the Applicability of Structural Criteria in Complexity Management. In: Culley, S. J.; Hicks, B. J.; McAlloone, T. C.; Howard, T. J.; Lindemann, U. (Eds.): Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 4: Product and Systems Design, Lyngby/Copenhagen, Denmark, 15.-19.08.2011, pp. 21-32. ISBN: 978-1-904670-24-7.

[BIEDERMANN & LINDEMANN 2011C]

Biedermann, W.; Lindemann, U.: Prediction of Communication Structures Based on Product Structures. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on

visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USA, 14-15 September 2011, pp. 291-300. ISBN 978-3446-43037-2.

[BIEDERMANN & LINDEMANN 2012]

Biedermann, W.; Lindemann, U.: Correlation of Structural Characteristics of Product Design Structure Matrices. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojcetic, N. (eds.): Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia, Vol. 3, pp. 1657-1666. ISBN 978-953-7738-20-4.

[BIEDERMANN ET AL. 2007]

Biedermann, W.; Maurer, M.; Lindemann, U.: The Multiple-Domain-Approach and Cost Attributes. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): Proceedings of the 9th International DSM Conference, Munich, Germany, 16.-18.10.2007, pp. 287-295. ISBN 3-8322-6641-0.

[BIEDERMANN ET AL. 2010]

Biedermann, W.; Strelkow, B.; Karl, F.; Lindemann, U.; Zaeh, M. F.: Reducing Data Acquisition Effort by Hierarchical System Modeling. In: Wynn, D. C.; Kreimeyer, M.; Eben, K.; Maurer, M.; Lindemann, U.; Clarkson, J. (eds.): Managing Complexity by Modeling Dependencies – Proceedings of the 12th International DSM Conference Cambridge, UK, 22 and 23 July 2010, pp. 309-318. ISBN: 978-3-446-42473-9.

[BIEDERMANN ET AL. 2011]

Biedermann, W.; Diepold, K.; Lindemann, U.: Entscheidungsstrategien zur Festlegung und Priorisierung von Änderungen. In: Maurer, M.; Schulze, S.-O. (eds.): Tag des Systems Engineering – Komplexe Herausforderungen meistern, Hamburg, 9.-11. November 2011, pp. 141-150. ISBN: 978-3-446-43041-9.

[BIEDERMANN ET AL. 2012]

Biedermann W., Diepold K. J., Lindemann U., Lohmann B.: Delimiting Structural and Dynamical System Analysis in Engineering Management. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojcetic, N. (eds.): Proceedings of DESIGN 2012, the 12th International Design Conference, Dubrovnik, Croatia, Vol. 3, pp. 1649-1656. ISBN 978-953-7738-20-4.

[BLEES & KRAUSE 2008]

Blees, C.; Krause, D.: On the Development of Modular Product Structures: A Differentiated Approach. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojcetic, N. (eds.): Proceedings DESIGN 2008, the 10th International Design Conference, Dubrovnik, Croatia, pp. 301-308. ISBN 978-953-6313-89-1.

[BLESSING & CHAKRABATI 2009]

Blessing, L.T.M., Chakrabati, A.: DRM, a Design Research Methodology. Springer 2009. ISBN 978-1-84882-587-1.

[BLOEBAUM & ENGLISH 1999]

Bloebaum, C.L.; English, K.: Using a Virtual DSM to improve Coupling Selection for Suspension in Multidisciplinary Design Optimization. In: Yassine, A.; Eppinger, S.; Whitney, D. (eds.): 1st Design Structure Matrix Workshop, 27 and 28 September 1999, Cambridge, MA. Massachusetts Institute of Technology 1999.

[BOHM ET AL. 2005]

Bohm, M.R.; Stone, R.B.; Szykman, S.: Enhancing Virtual Product Representations for Advanced Design Repository Systems. In: *Journal of Computing and Information Science in Engineering*, Vol. 5, Nr. 4, 2005, pp. 360-372. DOI:10.1115/1.1884618

[BONEAU 1960]

Boneau, C.A.: The effects of violations of assumptions underlying the t test. In: *Psychological Bulletin*, Vol. 57, No. 1, pp. 49–64.

[BONGULIEMI ET AL. 2001]

Bongulielmi, L.; Henseler, P.; Puls, C.; Meier, M.: The K- & V-Matrix Method - An Approach in Analysis and Description of Variant Products. In: Culley, S.; Duffy, A.; McMahon, C.; Wallace, K. (eds.): *Proceedings of ICED 01*, Glasgow, UK, 21.-23.08.2001. ISBN 1-86058-356-3.

[BRADY 2002]

Brady, T.K.: Utilization of Dependency Structure Matrix Analysis to Assess Implementation of NASA's Complex Technical Projects. Master Thesis Massachusetts Institute of Technology, 2002.

[BRAHA & BAR-YAM 2007]

Braha, D.; Bar-Yam, Y.: The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results. In: *Management Science*, Vol. 53, No. 7, pp. 1127-1145, 2007.

[BRAUN & DEUBZER 2007]

Braun, T.; Deubzer, F.: New Variant Management Using Multiple-Domain Mapping. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): *Proceedings of the 9th International DSM Conference*, Munich, Germany, 16.-18.10.2007, pp. 363-372. ISBN 3-8322-6641-0.

[BRAUN & LINDEMANN 2007]

Braun, S.C.; Lindemann, U.: A Multilayer Approach for Early Cost Estimation of Mechatronical Products. In: Bocquet, J.-C. (ed.): *Proceedings of ICED 2007*, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007, No. 187. ISBN 1-904670-02-4.

[BRØNDUM & ZHU 2010]

Brøndum, J.; Zhu, L.M.: Towards an architectural viewpoint for systems of software intensive systems. In: *SHARK '10 Proceedings of the 2010 ICSE Workshop on Sharing and Reusing*

Architectural Knowledge, Cape Town, South Africa, May 2-8, 2010 of ICSE International Conference on Software Engineering, pp. 60-63. ISBN: 978-1-60558-967-1.

[BROWNING & YASSINE 2010A]

Browning, T.R.; Yassine, A.A.: A Random Generator of Resource-Constrained Multi-Project Network Problems. In: *Journal of Scheduling*, Vol. 13, No. 2, pp. 143-161, 2010. DOI: 10.1007/s10951-009-0131-y

[BROWNING & YASSINE 2010B]

Browning, T.R.; Yassine, A.A.: Resource-Constrained Multi-Project Scheduling: Priority Rule Performance Revisited. In: *International Journal of Production Economics*, Vol. 126, No. 2, pp. 212-228, 2010. DOI: 10.1016/j.ijpe.2010.03.009

[BROWNING 2001]

Browning, T.R.: 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, 2001.

[BHUSHAN 2005]

Bhushan, N.: Robust Inventive Software Design (RISD): A Framework Combining Design Structure Matrix (DSM), Analytic Hierarchy Process (AHP) and TRIZ. In: *7th International Design Structure Matrix Conference*, Seattle, Washington, USA, October 4-6. The Boeing Company 2005.

[CAI & HUYNH 2007]

Cai, Y.F.; Huynh, S.: An Evolution Model for Software Modularity Assessment. In: *Fifth International Workshop on Software Quality (WoSQ'07: ICSE Workshops 2007)*, Minneapolis, Minnesota, May 20-May 26. ISBN: 0-7695-2959-3.

[CALLAHAN 2009]

Callahan, S.: Information Models for Complex Product Designs. In: Kreimeyer, M.; Maier, J.; Fadel, G.; Lindemann, U. (eds.): *Proceedings of the 11th International DSM Conference*, Greenville, SC, USA, 12.-13.10.2009. ISBN 978-3-446-42194-3.

[CAMI & DEO 2008]

Cami, A.; Deo, N.: Techniques for analysing dynamic random graph models of web-like networks: An overview. In: *Networks*, Vol. 51, No. 4, pp. 211-255, 2008.

[CASTRO 2010]

Castro, J.N.L.: *Individuals in Product Development: Interactions with Teams and Products*. PhD Thesis Massachusetts Institute of Technology, 2010.

[CHENG 2007]

Cheng, C.-J.: *Applying QFD and DSM for Product Platform Development and Evaluation*. Master Thesis Institute of Industrial Management, National Central University.

[CHIRIAC ET AL. 2011A]

Chiriac, N.; Hölttä-Otto, K.; Lysy, D.; Suh, E.S.: Three Approaches to Complex System Decomposition. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USE, 14-15 September 2011, pp. 11-25. ISBN 978-3446-43037-2.

[CHIRIAC ET AL. 2011B]

Chiriac, N.; Hölttä-Otto, K.; Lysy, D.; Suh, E.S.: Level of Modularity and Different Levels of System Granularity. In: Journal of Mechanical Design, Vol. 133, 101007, 2011.

[CHOW 1996]

Chow, S.L.: Statistical Significance: Rationale, Validity and Utility. Sage Publications Ltd, ISBN: 978-0-7619-5205-3.

[CHUA & ASLAM HOSSAIN 2012]

Chua, D.K.H.; Aslam Hossain, M.: Predicting Change Propagation and Impact on Design Schedule Due to External Changes. In: IEEE Transactions on Engineering Management, Vol. 59, No. 3, pp. 483-493, 2012.

[CLARKSON ET AL. 2004]

Clarkson, P.J.; Simons, C.; Eckert, C.: Predicting Change Propagation in Complex Design. In: Journal of Mechanical Design, Vol. 126, pp. 788-797, 2004.

[CONWAY 1968]

Conway, M.E.: How Do Committees Invent? In: Datamation, Vol. 14, No. 5, pp. 28-31, 1968.

[CRAWLEY ET AL. 2004]

Crawley, E.; de Weck, O.; Eppinger, S.; Magee, C.; Moses, J.; Seering, W.; Schindall, J.; Wallace, D.; Whitney, D.: The Influence of Architecture in Engineering Systems. Engineering Systems Monograph, Engineering Systems Monograph, Massachusetts Institute of Technology.

[DALEIDEN 1999]

Daleiden, S.A.: Exploring Design Drivers and Management Issues Using DSM. In: Yassine, A.; Eppinger, S.; Whitney, D. (eds.): 1st Design Structure Matrix Workshop, 27 and 28 September 1999, Cambridge, MA. Massachusetts Institute of Technology 1999.

[DANILOVIC & BROWNING 2004]

Danilovic, M.; Browning, T.R.: A formal approach for domain mapping matrices (DMM) to complement design structure matrices (DSM). In: Clarkson, P.J. (ed.): 6th International Design Structure Matrix Conference, Cambridge, UK, 12 - 14 September 2004.

[DE SOUZA & REDMILES 2005]

de Souza, C.; Redmiles, D.: An Interdisciplinary Perspective on Interdependencies. ISR Technical Report # UCI-ISR-05-7. Institute for Software Research, University of California, Irvine, 2005.

[DE WECK & SUH 2006]

de Weck, O.; Suh, E.S.: Flexible Product Platforms: Framework and Case Study. In: Proceedings of IDETC/CIE 2006 ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 10-13, 2006, Philadelphia, Pennsylvania, USA, DETC2006-99163.

[DE WECK 2007]

de Weck, O.: On the Role of DSM in Designing Systems and Products for Changeability. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): Proceedings of the 9th International DSM Conference, Munich, Germany, 16.-18.10.2007, pp. 311-323. ISBN 3-8322-6641-0.

[DE WECK ET AL. 2012]

de Weck, O.; Ross, A.M.; Rhodes, D.H.: Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities). In: Third International Engineering Systems Symposium, CESUN 2012, Delft University of Technology, 18-20 June 2012.

[DEMOLY ET AL. 2012]

Demoly, F.; Yan, X.-T., Eynard, B.; Gomes, S.; Kiritsis, D.: Integrated product relationships management: a model to enable concurrent product design and assembly sequence planning. In: Journal of Engineering Design, Vol. 23, No. 7, pp. 544-561, 2012. DOI: 10.1080/09544828.2011.629317

[DENKER 2006]

Denker, S.: DSMs for Dummies. In: 8th International Design Structure Matrix Conference, Seattle, Washington, USA, October 24-26. The Boeing Company 2006.

[DEO & PANG 1984]

Deo, N.; Pang, C.-Y.: Shortest-Path Algorithms: Taxonomy and Annotation. In: Networks, Vol. 14, pp. 275-323, 1984.

[DEUBZER ET AL. 2008]

Deubzer, F.; Braun, T.; Maurer, M.; Lindemann, U.: Applying the Multiple Domain Mapping Approach to Variant Management. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojctetic, N. (eds.): Proceedings DESIGN 2008, the 10th International Design Conference, Dubrovnik, Croatia, pp. 335-342. ISBN 978-953-6313-89-1.

[DIAZ-GARCIA 2009]

Diaz Garcia, J.A.: Network Analysis of Technical and Organizational Configurations: Using an Alignment Approach to Enhance Product Development Performance. Master Thesis Massachusetts Institute of Technology, 2009.

[DIEPOLD ET AL. 2010A]

Diepold, K.J.; Biedermann, W.; Eben, K. G. M.; Kortler, S.; Lohmann, B.; Lindemann, U.: Combining Structural Complexity Management and Hybrid Dynamical System Modelling. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojctetic, N. (eds.): Proceedings of DESIGN 2010,

the 11th International Design Conference, Dubrovnik, Croatia, pp. 1045-1054. ISBN 978-953-7738-03-7.

[DIEPOLD ET AL. 2010B]

Diepold, K.J.; Winkler, F.J.; Lohmann, B.: Systematical Hybrid State Modelling of Complex Dynamical Systems: The Quad-I/HS Framework. In: Journal of Mathematical and Computer Modelling of Dynamical Systems, Vol. 16, No. 4, pp.347-371, 2010.

[DO & CARIGNAN 2005]

Do, S.-H.; Carignan, J.: Software Dependency Analysis using DSM and UML Models. In: 7th International Design Structure Matrix Conference, Seattle, Washington, USA, October 4-6. The Boeing Company 2005.

[EBEN ET AL. 2008]

Eben, K.G.M.; Biedermann, W.; Lindemann, U.: Modeling Structural Change over Time - Requirements and First Methods. In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (eds.): Proceedings of the 10th International DSM Conference, Stockholm, Sweden, 11.-12.11.2008, page 15-24. ISBN: 978-3-446-41825-7.

[EBEN ET AL. 2010]

Eben, K.G.M.; Daniliidis, C.; Lindemann, U.: Interrelating and Prioritising Requirements on Multiply Hierarchy Levels. In: Marjanović, D.; Štorga, M.; Pavković, N.; Bojčetić, N. (eds.): Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik - Croatia, May 17 - 20, 2010, page 1055-1064. ISBN: 978-953-7738-03-7.

[ECKERT ET AL. 2004]

Eckert, C.M.; Clarkson, P.J.; Zanker: Change and customisation in complex engineering domains. In: Research in Engineering Design, Vol. 15, No. 1, pp. 1-21, 2004

[EICHINGER ET AL. 2006]

Eichinger, M.; Maurer, M.; Pulm, U.; Lindemann, U.: Extending Design Structure Matrices and Domain Mapping Matrices by Multiple Design Structure Matrices. In: Proceedings of ESDA2006 8th Biennial ASME Conference on Engineering Systems Design and Analysis, July 4-7, 2006, Torino, Italy, ESDA2006-95266.

[EINÖGG 2009]

Einögg, F.: Netzwerk-FMEA: Methodik und Anwendung. Student thesis, Institute of Product Development, Technische Universität München, 2009.

[EPPINGER & BROWNING 2012]

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

[EPPINGER 2009]

Eppinger, S.: Integrating the Product, Process, and Organization Views of Complex System Development. In: Kreimeyer, M.; Maier, J.; Fadel, G.; Lindemann, U. (eds.): Proceedings of the

11th International DSM Conference, Greenville, SC, 12 - 13 October 2009 , pp. 7-8. ISBN: 978-3-446-42194-3

[FLANAGAN 2007]

Flanagan, T.: Applying DSM in the Energy Sector: Practical Problems and Insights from Industry. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): Proceedings of the 9th International DSM Conference, Munich, Germany, 16.-18.10.2007, pp. 383-392. ISBN 3-8322-6641-0.

[FREEMAN 1978]

Freeman, L.C.: Centrality in Social Networks - Conceptual Clarification. In: Social Networks, Vol. 1, pp. 215-239, 1978.

[GERSHENSON ET AL. 2004]

Gershenson, J.K.; Prasad, G.J.; Zhang, Y.: Product modularity: measures and design methods. In: Journal of Engineering Design, Vol. 15, Nr. 1, pp. 33-51, 2004.

[GERSHENSON ET AL. 2007]

Gershenson, J.K.; Khadke, K.N., Lai, X.X.: A Research Roadmap for Product Family Design. In: Bocquet, J.-C. (ed.): Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007, No. 370. ISBN 1-904670-02-4.

[GIFFIN 2007]

Giffin, M.L.: Change Propagation in Large Technical Systems. Master Thesis, Massachusetts Institute of Technology, 2007.

[GIFFIN ET AL. 2009]

Giffin, M.; de Weck, O.; Bounova, G.; Keller, R.; Eckert, C.; Clarkson, P.J.: Change Propagation Analysis in Complex Technical Systems. In: Journal of Mechanical Design, Vol. 131, 081001, 2009.

[GOKPINAR ET AL. 2010]

Gokpinar, B.; Hopp, W.J.; Iravani, S.M.R.: The Impact of Misalignment of Organizational Structure and Product Architecture on Quality in Complex Product Development. In: Management Science, Vol. 56, No. 3, pp. 468-484, 2010.

[GREINER ET AL. 2007]

Greiner, M.; Warga, J.; Braun, T.: Complexity Management Using Multiple-Domain Mapping - Development of High Pressure Pumps for Common Rail Systems. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): Proceedings of the 9th International DSM Conference, Munich, Germany, 16.-18.10.2007, pp. 261-270. ISBN 3-8322-6641-0.

[GROSS & YELLEN 2003]

Gross, J.L.; Yellen, J.: Handbook of Graph Theory. CRC Press 2003. ISBN 978-1584880905.

[GU ET AL. 2012]

Gu, C.-C.; Hu, J.; Peng, Y.-H.; Li, S.: FCBS model for functional knowledge representation in conceptual design. In: *Journal of Engineering Design*, Vol. 23, No. 8, pp. 577-596, 2012.

[GÜNTHER 2011]

Günther, C.: Kennzahlensystem zur Bewertung von Rekonfigurationen an Betriebsmitteln. Student thesis, Institute for Machine Tools and Industrial Management, Technische Universität München, 2011.

[GUTIERREZ 1998]

Gutierrez Fernandez, C.I.: Integration Analysis of Product Architecture to Support Effective Team Co-Location. Master Thesis, Massachusetts Institute of Technology, 1998.

[HAMRAZ ET AL. 2012]

Hamraz, B.; Caldwell, N.H.M.; Clarkson, P.J.: A multidomain engineering change propagation model to support uncertainty reduction and risk management in design. In: *Journal of Mechanical Design*, Vol. 134, No. 10, 100905.01-14, 2012.

[HARARY ET AL. 1965]

Harary, F.; Norman, R.Z.; Cartwright, D.: *Structural Models - An Introduction to the Theory of Directed Graphs*. Wiley 1965.

[HARTIGAN 1975]

Hartigan, J.A.: *Clustering Algorithms*. Wiley 1975.

[HELMER ET AL. 2010]

Helmer, R.; Yassine, A.; Meier, C.: Systematic module and interface definition using component design structure matrix. In: *Journal of Engineering Design*, Vol. 21, No. 6, pp. 647-675, 2010.

[HELMS & SHEA 2011]

Helms, B.; Shea, K.: Computational Synthesis of Product Architectures Based on Object-Oriented Graph Grammars. In: *Journal of Mechanical Design*, Vol. 134, 021008, 2012.

[HELO & HILMOLA 2003]

Helo, P.; Hilmola, O.-P.: DSM based Value Analysis of Product Platforms. In: Clarkson, P.J. (ed.): *5th Design Structure Matrix Workshop*, Cambridge, UK, 22 and 23 October 2003. University of Cambridge, Cambridge, UK, 2003.

[HERFELD 2008]

Herfeld, U.: From the Real Product to its Abstract Architecture and back again. In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (eds.): *Proceedings of the 10th International DSM Conference*, Stockholm, Sweden, 11.-12.11.2008, page 15-24. ISBN: 978-3-446-41825-7.

[HERFELD ET AL. 2007]

Herfeld, U.; Fürst, F.; Braun, T.: Managing Complexity in Automotive Safety Development. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): *Proceedings of*

the 9th International DSM Conference, Munich, Germany, 16.-18.10.2007, pp. 271-283. ISBN 3-8322-6641-0.

[HOFSTETTER & DE WECK 2007]

Hofstetter, W.K.; Wooster, P.D.; de Weck, O.L.; Crawley, E.F.: The System Overlap Matrix - A Method and Tool for the Systematic Identification of Commonality Opportunities in Complex Technical Systems. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): Proceedings of the 9th International DSM Conference, Munich, Germany, 16.-18.10.2007, pp. 215-224. ISBN 3-8322-6641-0.

[HOLMQVIST & PERSSON 2003]

Holmqvist, T.K.P.; Persson, M.L.: Analysis and Improvement of Product Modularization Methods: Their Ability to Deal with Complex Products. In: Systems Engineering, Vol. 6, No. 3, pp. 195-209, 2003.

[HÖLTTA-OTTO & DE WECK 2007]

Höltkä-Otto, K.; de Weck, O.: Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints. In: Concurrent Engineering: Research and Applications, Vol. 15, No. 2, pp. 113-126, 2007.

[HOSSAIN & ZULKARNINE 2011]

Hossain, S.; Zulkarnine, A.T.: Design Structure of Scientific Software - A Case Study. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USE, 14-15 September 2011, pp. 129-141. ISBN 978-3446-43037-2.

[INCOSE 2011]

INCOSE: Systems Engineering Handbook v3.2.1. INCOSE 2011.

[JARRATT ET AL. 2011]

Jarratt, T.A.W.; Eckert, C.M.; Caldwell, N.H.M.; Clarkson, P.J.: Engineering change: An overview and perspective on the literature. In: Research in Engineering Design, Vol. 22, No. 2, pp. 103-124, 2011

[JOHANNESSON & SÖDERBERG 2000]

Johannesson, H.; Söderberg, R.: Structure and Matrix Models for Tolerance Analysis from Configuration to Detail Design. In: Research in Engineering Design, Vol. 12, pp. 112-125, 2000.

[KALLIGEROS ET AL. 2006]

Kalligeros, K.; de Weck, O.; de Neufville, R.: Platform identification using Design Structure Matrices. In: 2006 Sixteenth Annual International Symposium of INCOSE "Systems Engineering: Shining Light on the Tough Issues", Orlando, FL, 09.-13.07.2006, 2006. ISBN: 0-9720562-4-6.

[KELLER ET AL. 2005]

Keller, R.; Eckert, C.M.; Clarkson, P.J.: Multiple Views to Support Engineering Change Management for Complex Products. In: Proceedings of the Third International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV'05), 2005. ISBN 0-7695-2396-X.

[KELLER ET AL. 2006]

Keller, R.; Eckert, C.M.; Clarkson, P.J.: Computing Change Propagation in DSMs. In: 8th International Design Structure Matrix Conference, Seattle, Washington, USA, October 24-26. The Boeing Company 2006.

[KELLER ET AL. 2009]

Keller, R.; Eckert, C.M.; Clarkson, P.J.: Using an engineering change methodology to support conceptual design. In: Journal of Engineering Design, Vol. 20, No. 6, pp. 571-587, 2009.

[KESPER 2013]

Kesper, H.: Gestaltung von Produktvariantenspektren mittels matrixbasierter Methoden. Dissertation, Technische Universität München.

[KISSEL ET AL. 2011]

Kissel, M.P.; Hellenbrand, D.; Lindemann, U.: A Methodology to Evaluate the Structural Robustness of Product Concepts. In: Culley, S.J.; Hicks, B.J.; McAlone, T.C.; Howard, T.J.; Dong, A. (eds.): Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 9: Design Methods and Tools pt. 1, Lyngby/Copenhagen, Denmark, 15.-19.08.2011, pp. 215-225. ISBN 978-1-904670-29-2.

[KOH & CLARKSON 2009]

Koh, E.C.Y.; Clarkson, P.J.: A Modelling Method to Manage Change Propagation. In: Norell Bergendahl, M.; Grimheden, M.; Leifer, L.; Skogstad, P.; Lindemann, U. (eds.): Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 1, Design Processes, Palo Alto, CA, USA, 24.-27.08.2009, pp. 253-264. ISBN: 978-1-904670-05-6.

[KOH ET AL. 2007]

Koh, E.C.Y.; Keller, R.; Eckert, C.M.; Clarkson, P.J.: Component Classification: A Change Perspective. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): Proceedings of the 9th International DSM Conference, Munich, Germany, 16.-18.10.2007, pp. 337-348. ISBN 3-8322-6641-0.

[KOH ET AL. 2008]

Koh, E.C.Y.; Keller, R.; Eckert, C.M.; Clarkson, P.J.: Influence of Feature Change Propagation on Product Attributes in Concept Selection. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojetic, N. (eds.): Proceedings DESIGN 2008, the 10th International Design Conference, Dubrovnik, Croatia, pp.157-166. ISBN 978-953-6313-89-1.

[KOH ET AL. 2009]

Koh, E.C.Y.; Caldwell, N.H.M.; Clarkson, P.J.: Using a Matrix-Based Approach to Model Change Propagation. In: Kreimeyer, M.; Maier, J.; Fadel, G.; Lindemann, U. (eds.):

Proceedings of the 11th International DSM Conference, Greenville, SC, 12 - 13 October 2009 , pp. 271-284. ISBN: 978-3-446-42194-3.

[KOH ET AL. 2012]

Koh, E.C.Y.; Caldwell, N.H.M.; Clarkson, P.J.: A method to assess the effects of engineering change propagation. In: *Research in Engineering Design*, Vol. 23, No. 4, pp. 329-351, 2012.

[KONG 2011]

Kong, F.Y.: Weiterentwicklung einer Methodik zur Planung von Rekonfigurationen an Montagebetriebsmitteln. Student thesis, Institute for Machine Tools and Industrial Management, Technische Universität München, 2011.

[KORTLER ET AL. 2009]

Kortler, S.; Kreimeyer, M.; Lindemann, U.: A Planarity-Based Complexity Metric. In: Norell Bergendahl, M.; Grimheden, M.; Leifer, L.; Skogstad, P.; Lindemann, U. (eds.): *Proceedings of ICED 09, the 17th International Conference on Engineering Design*, Vol. 6, Design Methods and Tools (pt. 2), Palo Alto, CA, USA, 24.-27.08.2009, pp. 31-42. ISBN: 978-1-904670-10-0.

[KREIMEYER & LINDEMANN 2011]

Kreimeyer, M.; Lindemann, U.: *Complexity metrics in engineering design*. Springer 2011.

[KREIMEYER ET AL. 2006]

Kreimeyer, M.; Herfeld, U.; Deubzer, F.; Dequindt, C.; Lindemann, U.: Function-Driven Product Design in Virtual Teams through Methodical Structuring of Requirements and Components. In: *Proceedings of ESDA2006 8th Biennial ASME Conference on Engineering Systems Design and Analysis*, July 4-7, 2006, Torino, Italy, ESDA2006-95380.

[KUSIAK & WANG 1995]

Kusiak, A.; Wang, J.T.: Dependency Analysis in Constraint Negotiation. In: *IEEE Transactions on System, Man and Cybernetics*, Vol. 25, No. 9, pp. 1301-1313, 1995.

[KUSIAK 1993]

Kusiak, A.: *Concurrent Engineering - Automation, Tools, and Techniques*. Wiley 1993. ISBN: 0-471-55492-8.

[KUSIAK 2002]

Kusiak, A.: Integrated product and process design: a modularity perspective. In: *Journal of Engineering Design*, Vol. 13, No. 3, pp. 223-231, 2002.

[KUSIAK 2007]

Kusiak, A.: From Product/Service Complexity Management to Innovation. In: Lindemann, U.; Danilovic, M.; Deubzer, F.; Maurer, M.; Kreimeyer, M. (eds.): *Proceedings of the 9th International DSM Conference*, Munich, Germany, 16.-18.10.2007. ISBN 3-8322-6641-0.

[KUSIAK 2008]

Kusiak, A.: Interface Structure Matrix for Analysis of Products and Processes. In: Proceedings of the 15th CIRP International Conference on Life Cycle Engineering, LCE 2008, The University of New South Wales, Sydney, Australia, March 2008, pp. 444-448.

[LAMANTIA 2006]

LaMantia, M.J.: Dependency Models as a Basis for Analysing Software Product Platform Modularity: A Case Study in Strategic Software Design Rationalization. Master Thesis, Massachusetts Institute of Technology, 2006.

[LANGER ET AL. 2010]

Langer, S.; Knoblinger, C.; Lindemann, U.: Analysis of Dynamic Changes and Iterations in the Development Process of an Electrically Powered Go-Kart. In: Marjanovic, D.; Storga, M.; Pavkovic, N.; Bojetic, N. (eds.): Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia, pp. 307-318. ISBN 978-953-7738-03-7.

[LAPP & GOLAY 1997]

Lapp, C.W.; Golay, M.W.: Modular design and construction techniques for nuclear power plants. In: Nuclear Engineering and Design, Vol. 172, pp. 327-349, 1997.

[LARSES 2005]

Larses, O.: Applying quantitative methods for architecture design of embedded automotive systems. In: 2005 Fifteenth Annual International Symposium of INCOSE, "Systems Engineering: Bridging Industry, Government, and Academia", Rochester, NY, 10.-14.07.2005. ISBN: 0-9720562-3-8.

[LAU 2006]

Lau, H.T.: A Java Library of Graph Algorithms and Optimization. CRC Press 2006.

[LI & CHEN 2008]

Li, S.; Chen, L.: A Unified Framework for Decomposition of Design Structure Matrix and Domain Mapping Matrix. In: Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008, August 3-6, 2008, Brooklyn, New York, USA, DETC2008-49542.

[LINDEMANN ET AL. 2009]

Lindemann, U.; Maurer, M.; Braun, T.: Structural Complexity Management - An Approach for the Field of Product Design. Springer 2009.

[LÜNING 2011]

Lüning, B.: Priorisierungsregeln für Änderungen - Evaluation mittels Simulation. Diploma Thesis, Institute of Product Development, Technische Universität München, 2011.

[MACCORMACK & RUSNAK 2004]

MacCormack, A.; Rusnak, J.: The application of DSMs to the analysis of complex software designs. In: Clarkson, P.J. (ed.): 6th International Design Structure Matrix Conference, Cambridge, UK, 12 - 14 September 2004. University of Cambridge 2004.

[MACCORMACK ET AL. 2006]

MacCormack, A.; Rusnak, J.; Baldwin, C.Y.: Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code. In: *Management Science*, Vol. 52, No. 7, pp. 1015-1030, 2006.

[MAGEE 2001]

Magee, C.: The Readiness of DSM methodology for widespread deployment. In: Eppinger, S.; Whitney, D.; Yassine, A. (eds.): *3rd Design Structure Matrix Workshop*, Cambridge, MA, 29 and 30 October 2001. Massachusetts Institute of Technology 2001.

[MAIER ET AL. 2007]

Maier, J.R.A.; Ezhilan, T.; Fadel, G.M., Summers, J.D.: A Hierarchical Requirements Modeling Scheme to Support Engineering Innovation. In: Bocquet, J.-C. (ed.): *Proceedings of ICED 2007, the 16th International Conference on Engineering Design*, Paris, France, 28.-31.07.2007, No.582. ISBN 1-904670-02-4.

[MALMQVIST 2002]

Malmqvist, J.: A Classification of Matrix-Based Methods for Product Modelling. In: Marjanovic, D. (ed.): *Proceedings of the 7th International Design Conference - DESIGN 2002*, Dubrovnik-Cavtat, Croatia, May 14-17, 2002, pp. 203-210. ISBN 953-6313-45-6.

[MALMSTRÖM & MALMQVIST 1998]

Malmström, J.; Malmqvist, J.: Trade Off Analysis in Product Structures: A Case Study at Celsius Aerotech. In: Andersson, K.; Persson, J.-G. (eds.): *Proceedings of NordDesign'98*, 26.-28.08.1998, Stockholm, Sweden Volume 1998, 15, pp. 187-196. Royal Institute of Technology, KTH, Department of Machine Design, 1998.

[MANTEUFFEL 2011]

Manteuffel, D.C.: Erstellung eines Kennzahlensystems zur Bewertung der Rekonfigurationsfähigkeit von Betriebsmitteln. Diploma thesis, Institute for Machine Tools and Industrial Management, Technische Universität München, 2011.

[MARTI 2007]

Marti, M.: Complexity Management: Optimizing Product Architecture of Industrial Products. Dissertation of the University of St. Gallen, Graduate School of Business Administration, Economics, Law and Social Sciences (HSG), 2007.

[MARTIN & ISHII 2002]

Martin, M.V.; Ishii, K.: Design for variety: developing standardized and modularized product platform architectures. In: *Research in Engineering Design*, Vol. 13, pp. 213-235, 2002.

[MATTHIESON & SUMMERS 2009]

Matthieson, J.; Summers, J.: Relational DSMs in Connectivity Complexity Measurement. In: Kreimeyer, M.; Maier, J.; Fadel, G.; Lindemann, U. (eds.): *Proceedings of the 11th International DSM Conference*, Greenville, SC, 12 - 13 October 2009, pp. 15-26. ISBN: 978-3-446-42194-3.

- [MAURER & LINDEMANN 2007]
Maurer, M.; Lindemann, U.: Facing Multi-Domain Complexity in Product Development. In: CiDaD Working Paper Series, Vol. 3, No. 1, pp. 1-12, 2007.
- [MAURER 2007]
Maurer, M.S.: Structural Awareness in Complex Product Design. Dissertation, Technische Universität München, 2007.
- [MAURER 2011]
Maurer, M.: Komplexitätsmanagement für die industrielle Praxis. Lecture Notes, Institute of Product Development, Technische Universität München, 2011.
- [McCORMICK ET AL. 1972]
McCormick, W.T. Jr.; Schweitzer, P.J.; White, T.W.: Problem Decomposition and Data Reorganization by a Clustering Technique. In: Operations Research, Vol. 20, No. 5, pp. 993-1009, 1972.
- [McLELLAN ET AL. 2009]
McLellan, J.M.; Maier, J.R.A.; Fadel, G.M.; Mocko, G.M.: A Method for Identifying Requirements Critical to Mass Reduction Using DSMs and DMMs. In: Kreimeyer, M.; Maier, J.; Fadel, G.; Lindemann, U. (eds.): Proceedings of the 11th International DSM Conference, Greenville, SC, 12 - 13 October 2009 , pp. 197-205. ISBN: 978-3-446-42194-3.
- [MILO ET AL. 2002]
Milo, R.; Shen-Orr, S.; Itzkovitz, S.; Kashtan, N.; Chklovskii, D; Alon, U.: Network Motifs: Simple Building Blocks of Complex Networks. In: Science, Vol. 298, No. 5594, pp. 824-827, 2002.
- [MOCKO ET AL. 2007]
Mocko, G.M.; Summers, J.D.; Fadel, G.M.; Teegavarapu, S.; Maier, J.R.A.; Ezhilan, T.: A Modelling Scheme for Capturing and Analysing Multi-Domain Design Information: A Hair Dryer Design Example. In: Bocquet, J.-C. (ed.): Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007, No. 570. ISBN 1-904670-02-4.
- [MORELLI ET AL 1995]
Morelli, M.D.; Eppinger, S.D.; Gulati, R.K.: Predicting Technical Communication in Product Development Organizations. In: IEEE Transactions on Engineering Management, Vol. 42, No. 3, pp. 215-222, 1995.
- [NORD ET AL. 2011]
Nord, R.L.; Ozkaya, I.; Brown, N.; Sangwan, R.S.: Modeling Architectural Dependencies to Support Software Release Planning. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USA, 14-15 September 2011, pp. 159-171. ISBN 978-3446-43037-2.

[PARNAS 1972]

Parnas, D.L.: On the Criteria to be Used in Decomposing Systems into Modules. In: Communications of the ACM, Vol. 15, No. 12, pp. 1053-1058, 1972.

[PASQUAL & DE WECK 2012]

Pasqual, M.C.; de Weck, O.L.: Multilayer Network Model for Analysis and Management of Change Propagation. In: Research in Engineering Design, Vol. 23, No. 4, pp. 305-328, 2012.

[PASQUAL 2010]

Pasqual, M.C.: Multilayer Network Model for Analysis and Management of Change Propagation. Master Thesis, Massachusetts Institute of Technology, 2010.

[PIMMLER & EPPINGER 1994]

Pimmler, T.U.; Eppinger, S.D.: Integration Analysis of Product Decompositions. In: Hight, T.K.; Mistree, F. (eds.): Design theory and methodology, DTM '94. ISBN: 0-7918-1282-0.

[RECHTIN 1991]

Rechtin, E.: Systems architecting: Creating and building complex systems. Prentice Hall 1991.

[ROCCO ET AL. 2011]

Rocco, C.; de Napoli, L.; Rizzuti, S.: A Proposal for an Augmented DSM to Assess Product Sustainability. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USA, 14-15 September 2011, pp. 45-57. ISBN 978-3446-43037-2.

[ROOSMALEN 2008]

Roosmalen, H.: Introduction of Software Related DSMs to Software Engineers (A Case Study). In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (eds.): Proceedings of the 10th International DSM Conference, Stockholm, Sweden, 11.-12.11.2008, page 3-13. ISBN: 978-3-446-41825-7.

[ROSS ET AL. 2008]

Ross, A.M.; Rhodes, D.H.; Hastings, D.E.: Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value. In: Systems Engineering, Vol. 11, No. 3, pp. 246-262, 2008.

[RUSHTON ET AL. 2002]

Rushton, G.; Zakarian, A.; Grigoryan, T.: Algorithms and Software for Development of Modular Architectures. In: 12th Annual International Symposium of INCOSE "Engineering 21st Century Systems: Problem Solving Through Structured Thinking", Las Vegas, NV, 28.07.-01.08.2002. INCOSE 2002.

[SANGAL & WALDMAN 2005]

Sangal, N.; Waldman, F.: Dependency Models to Manage Software Architecture, In: CROSSTALK The Journal of Defense Software Engineering, pp. 8-12, 2005.

[SANGAL ET AL. 2005]

Sangal, N.; Jordan, E.; Sinha, V.; Jackson, D.: Using Dependency Models to Manage Complex Software Architecture. In: 20th Annual ACM Conference on Object-Oriented Programming, Systems, Languages and Applications, OOPSLA '05, San Diego, CA, 16.-20.10.2005 , pp. 167-176. ISBN: 1-59593-031-0.

[SCHMITZ ET AL. 2011]

Schmitz, S.; Wynn, D.C.; Biedermann, W.; Lindemann, U.; Clarkson, P.J.: Improving Data Quality in DSM Modelling: A Structural Comparison Approach. In: Culley, S.J.; Hicks, B.J.; McAlone, T.C.; Howard, T.J.; Lindemann, U. (eds.): Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design, Vol. 4: Product and Systems Design, Lyngby/Copenhagen, Denmark, 15.-19.08.2011, pp. 369-380, 2011. ISBN 978-1-904670-24-7.

[SHANKAR ET AL. 2012]

Shankar, P.; Morkos, B.; Summers, J.D.: Reasons for change propagation: a case study in an automotive OEM. In: Research in Engineering Design, Vol. 23, No. 4, pp. 291-303, 2012.

[SHARMAN 2001]

Sharman, D.M.: Valuing Architecture for Strategic Purposes - Comments on Applying the Dependency Structure Matrix with Real Options Theory. Master Thesis, Massachusetts Institute of Technology, 2002.

[SHARMAN ET AL. 2002A]

Sharman, D.M.; Yassine, A.A.; Carlile, P.: Architectural Optimisation Using Real Options Theory and Dependency Structure Matrices. In: Proceedings of DETC '02 ASME 2002, International Design Engineering Technical Conferences, 28th Design Automation Conference, Montreal, Canada, September 29-October 2, 2002, DETC2002/DAC-34119.

[SHARMAN ET AL. 2002B]

Sharman, D.M.; Yassine, A.A.; Carlile, P.: Characterising Modular Architectures. In: Proceedings of DETC '02, ASME 2002 International Design Engineering Technical Conferences, Design Theory & Methodology Conference, Montreal, Canada, September 29-October 2, 2002, DETC2002/DTM-34024.

[SMALING & DE WECK 2007]

Smaling, R.; de Weck, O.: Assessing Risks and Opportunities of Technology Infusion in System Design. In: Systems Engineering, Vol. 10, No. 1, pp. 1-25, 2007.

[SMALING 2005]

Smaling, R.: System Architecture Analysis and Selection Under Uncertainty. PhD Thesis, Massachusetts Institute of Technology, 2005.

[SOSA 2007]

Sosa, M.E.: Aligning Process, Product, and Organizational Architectures in Software Development. In: Bocquet, J.-C. (ed.): Proceedings of ICED 2007, the 16th International

Conference on Engineering Design, Paris, France, 28.-31.07.2007, No. 209. ISBN 1-904670-02-4.

[SOSA 2008]

Sosa, M.E.: A structured approach to predicting and managing technical interactions in software development. *Research in Engineering Design*, Vol. 19, pp. 47-70, 2008.

[SOSA ET AL. 2003]

Sosa, M.E.; Eppinger, S.D.; Rowles, C.M.: Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions. In: *Journal of Mechanical Design*, Vol. 125, pp. 240-252, 2003.

[SOSA ET AL. 2004]

Sosa, M.E.; Eppinger, S.D.; Rowles, C.M.: The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. In: *Management Science*, Vol. 50, No. 12, pp. 1674-1689, 2004.

[SOSA ET AL. 2007A]

Sosa, M.E.; Eppinger, S.D.; Rowles, C.M.: A Network Approach to Define Modularity of Components in Complex Products. In: *Journal of Mechanical Design*, Vol. 129, No. 11, pp. 1118-1129, 2007.

[SOSA ET AL. 2007B]

Sosa, M.E.; Browning, T.R.; Mihm, J.: Studying the Dynamics of the Architecture of Software Products. In: *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2007*, September 4-7, 2007, Las Vegas, Nevada, USA, DETC2007-34761.

[SOSA ET AL. 2008]

Sosa, M.E.; Browning, T.R.; Mihm, J.: A Dynamic, DSM-based View of Software Architectures and Their Impact on Quality and Innovation. In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (eds.): *Proceedings of the 10th International DSM Conference*, Stockholm, Sweden, 11.-12.11.2008, page 313-325. ISBN: 978-3-446-41825-7.

[SOSA ET AL. 2011]

Sosa, M.; Mihm, J.; Browning, T.: Degree Distribution and Quality in Complex Engineered Systems. In: *Journal of Mechanical Design*, Vol. 133, 101008, 2011.

[STEIGER 1980]

Steiger, J.H.: Tests for comparing elements of a correlation matrix. In: *Psychological Bulletin*, Vol. 87, pp. 245-251, 1980.

[STEVA ET AL. 2006]

Steva, E.D.; Rice, E.N.; Marion, T.J., Simpson, T.W., Stone, R.B.: Two Methodologies for Identifying Product Platform Elements within an Existing Set of Products. In: *Proceedings of IDETC/CIE 2006, ASME 2006 International Design Engineering Technical Conferences &*

Computers and Information in Engineering Conference, September 10-13, 2006, Philadelphia, Pennsylvania, USA, DETC2006-99234.

[STEWART 1981]

Steward, D.V.: The design structure system: A method for managing the design of complex systems. In: IEEE Transactions on Engineering Management, Vol. 28, No. 1, pp. 71-74, 1981.

[STRELKOW 2010]

Strelkow, B. Strukturmodellierung und Strukturanalyse zur Bestimmung der Anpassungsfähigkeit einer Produktionsanlage. Student thesis, Institute of Product Development, Technische Universität München, 2010.

[STROGATZ 2001]

Strogatz, S. H.: Exploring complex networks. In: Nature, Vol. 410, No. 6825, pp. 268-276, 2001.

[SUH ET AL. 2007]

Suh, E.S.; de Weck, O.L.; Chang, D.: Flexible product platforms: framework and case study. In: Research in Engineering Design, Vol. 18, No. 2, pp. 67-89, 2007.

[SUH ET AL. 2008]

Suh, E.S.; Furst, M.R.; Mihalyov, K.J.; de Weck, O.L.: Technology Infusion: An Assessment Framework and Case Study. In: Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2008, August 3-6, 2008, Brooklyn, New York, USA, DETC2008-49860.

[SULLIVAN ET AL. 2001]

Sullivan, K.J.; Griswold, W.G.; Cai, Y.F.; Hallen, B.: The Structure and Value of Modularity in Software Design. In: Gruhn, V. (ed.): Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering, Vienna, Austria, 10.-14.09.2001, pp. 99-108. ISBN: 1-58113-390-1.

[SUMMERS & SHAH 2010]

Summers, J.D.; Shah, J.J.: Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability. In: Journal of Mechanical Design, Vol. 132, 021004, 2010.

[TESEON 2011]

Teseon GmbH: Example delivered with LOOMEOTM 2.5.0. 2011.

[TILSTRA ET AL. 2009]

Tilstra, A.H.; Campbell, M.I.; Wood, K.L.; Seepersad, C.C.: Comparing Matrix-Based and Graph-Based Representations for Product Design. In: Wynn, D. C.; Kreimeyer, M.; Eben, K.; Maurer, M.; Lindemann, U.; Clarkson, J. (eds.): Managing Complexity by Modeling Dependencies – Proceedings of the 12th International DSM Conference Cambridge, UK, 22 and 23 July 2010, pp. 195-206. ISBN: 978-3-446-42473-9.

[ULRICH 1995]

- Ulrich, K.: The role of product architecture in the manufacturing firm. In: *Research Policy*, Vol. 24, pp. 419-441, 1995.
- [VON LUXBURG 2007]
von Luxburg, U.: A Tutorial on Spectral Clustering. Max Planck Institute for Biological Cybernetics 2007.
- [WALDMAN 2006]
Waldman, F.: Industry Adoption of DSM for Software Engineering. In: 8th International Design Structure Matrix Conference, Seattle, Washington, USA, October 24-26. The Boeing Company, 2006.
- [WARFIELD 1999]
Warfield, J.N.: Enhancing Understanding in the Domain of Complexity through a Model Exchange Strategy. In: Yassine, A.; Eppinger, S.; Whitney, D. (eds.): 1st Design Structure Matrix Workshop, 27 and 28 September 1999, Cambridge, MA. Massachusetts Institute of Technology, Cambridge, MA, 1999.
- [WELCH 1947]
Welch, B.L.: The generalization of "Student's" problem when several different population variances are involved. In: *Biometrika* 3, Vol. 4, Nr. 1-2, pp. 28-35.
- [WHITNEY 2011]
Whitney, D.: Perspectives on the DSM. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USA, 14-15 September 2011. ISBN 978-3446-43037-2.
- [WILDS 2008]
Wilds, J.M.: A Methodology for Identifying Flexible Design Opportunities. Master Thesis, Massachusetts Institute of Technology, 2008.
- [WOLFRAM 1983]
Wolfram, S.: Statistical mechanics of cellular automata. In: *Reviews of Modern Physics*, Vol. 55, No. 3, pp. 601-644, 1983.
- [WYNN ET AL. 2009]
Wynn, D.C.; Nair, S.M.T; Clarkson, P.J.: The P3 Platform: An Approach and Software System for Developing Diagrammatic Model-Based Methods in Design Research. In: Norell Bergendahl, M.; Grimheden, M.; Leifer, L.; Skogstad, P.; Lindemann, U. (eds.): Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 1, Design Processes, Palo Alto, CA, USA, 24.-27.08.2009, pp. 559-570. ISBN: 978-1-904670-05-6.
- [YANG & DUAN 2012]
Yang, F.; Duan, G.-J.: Developing a parameter linkage-based method for searching change propagation paths. In: *Research in Engineering Design*, Vol. 23, No. 4, pp. 353-372, 2012.
- [YASSINE & WISSMAN 2007]

Yassine, A.A.; Wissman, L.A.: The Implications of Product Architecture on the Firm. In: Systems Engineering, Vol. 10, No. 2, pp. 118-137, 2007.

[YASSINE 2011]

Yassine, A.A.: Multi-Domain DSM: Simultaneous Optimization of Product, Process & People DSMs. In: Wynn, D.; Kreimeyer, M.; Eben, K.; Maurer, M.; Lindemann, U.; Clarkson, P.J. (eds.): Managing Complexity by Modelling Dependencies - Proceedings of the 12th International DSM Conference, Cambridge, UK, 22.-23.07.2010, pp. 319-332. ISBN: 978-3-446-42473-9.

[YU ET AL. 2003]

Yu, T.-L.; Yassine, A.A.; Goldberg, D.E.: A Genetic Algorithm for Developing Modular Product Architectures. In: Proceedings of DETC'03, ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, Illinois, USA, September 2-6, 2003, DETC2003/DTM-48647.

[YU ET AL. 2007]

Yu, T.-L.; Yassine, A.A.; Goldberg, D.E.: An information theoretic method for developing modular architectures using genetic algorithms. Research in Engineering Design, Vol. 18, pp. 91-109, 2007.

[ZACHARIAS & YASSINE 2008]

Zacharias, N.A.; Yassine, A.A.: Optimal platform investment for product family design. In: Journal of Intelligent Manufacturing, Vol. 19, No. 2, pp. 131-148, 2008.

[ZAEH ET AL. 2010]

Zaeh, M. F.; Reinhart, G.; Lindemann, U.; Karl, F., Biedermann, W.: Evaluating the Innovation Ability of Manufacturing Resources. In: World Academy of Science, Engineering and Technology, Vol. 72, 2010, pp. 142-148. ISSN 2010-376X.

[ZAEH ET AL. 2011]

Zaeh, M. F.; Reinhart, G.; Lindemann, U.; Karl, F., Biedermann, W.: DSM-Based Evaluation of Assembly Manufacturing Resources. In: Eppinger, S.D.; Maurer, M.; Eben, K.; Lindemann, U. (eds.): Invest on visualization – Proceedings of the 13th International DSM Conference, Cambridge, MA, USA, 14-15 September 2011, pp. 435-448. ISBN 978-3-446-43037-2.

[ZAKARIAN 2001]

Zakarian, A.; Rushton, G.J.: Development of Modular Electrical Systems. IEEE/ASME Transactions on Mechatronics, Vol. 6, No. 4, pp. 507-520, 2001.

[ZHANG ET AL. 2012]

Zhang, L.L.; Xu, Q.; Yu, Y.; Jiao, R.J.: Domain-based production configuration with constraint satisfaction. In: International Journal of Production Research, Vol. 50, No. 24, pp. 7149-7166, 2012.

11. Appendix

11.1 Subsets of graphs

Table 11-1 Definition of the subsets of graphs

Subset	Definition
Block	Set of nodes. Each pair of nodes is connected via at least two independent paths.
Clique	Set of nodes, which are directly connected to each other
Component	Set of nodes. Each pair of nodes is connected via at least one path.
Cycle	Set of edges, which form a loop.
Path	Set of edges, which form a string between two nodes.

11.2 Global structural metrics

Table 11-2 Definitions of the global metrics

Metric	Definition
Average clustering coefficient	Average relational density of node locality.
Average degree	Ratio between number of edges and number of nodes.
Average distance centrality	Average minimum distance to each node.
Average number of blocks per node	Average number of blocks per node
Average number of cliques per node	Average number of cliques per node
Average number of cycles per node	Average number of cycles per node
Average path centrality	Average percentage of shortest paths running across a node.
Average path length	Average of the distances between all pairs of nodes.
Number of blocks	Number of clusters connected to other clusters via one node.
Number of cliques	Number of cluster which are internally fully connected.
Number of components	Number of clusters each pair of node at least indirectly connected.
Number of cycles	Number of edge chains which form a loop.
Number of edges	Number of edges
Number of nodes	Number of nodes

11.3 Local structural metrics

Table 11-3 Definitions of the local metrics

Metric	Definition
Average distance to node	Average distance to node.
Blocks per node	Number of blocks per node.
Cliques per node	Number of cliques per node.
Clustering coefficient	Relational density of node locality.
Cycles per node	Number of cycles per node.
Degree	Number of neighboring nodes.
Distance centrality	Minimum distance to each node.
Maximum distance to node	Maximum distance to node.
Median distance to node	Median distance to node.
Path centrality	Percentage of shortest paths running across a node.

11.4 Relations by definition among structural analyses

Table 11-4 Implications of relations by definition among structural analyses

Relation	Implication
Inheritance	The child analysis has the same significance as its parent. As the child is more specific and fulfills more conditions its significance may be a special case of the parent's.
Composition	The significance of the composition analysis is an aggregation of the significance of its parts. The significances must not contradict each other. Part analyses of the same composition may not be related.
Derivation	The derived analysis is either a property of a subset of the network or an aggregation of metrics. Its significance is more general than the original analysis' or highlights the original's significance partially.

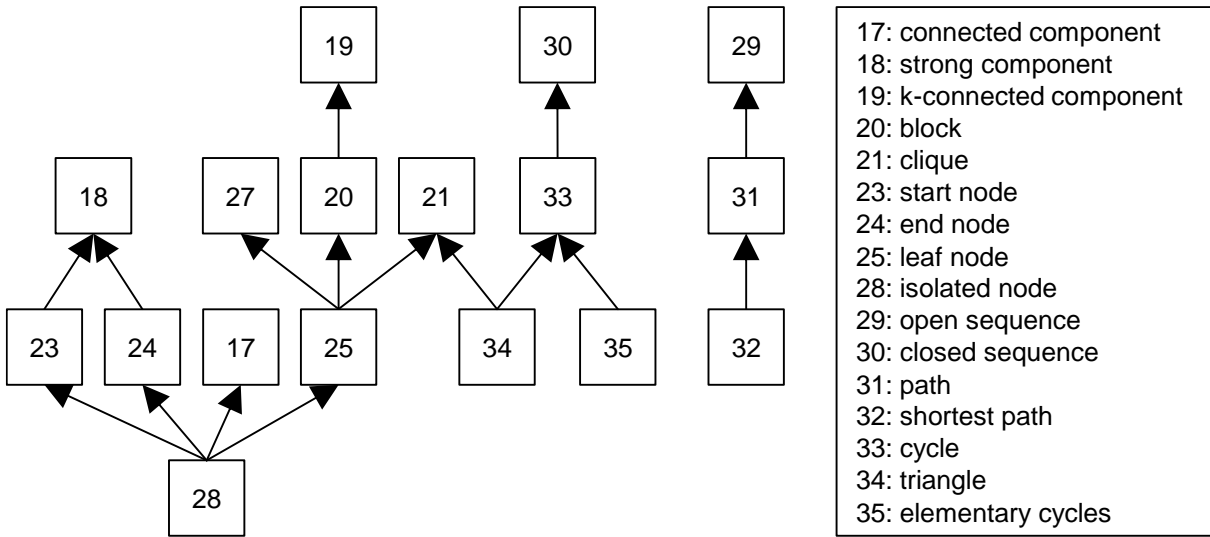
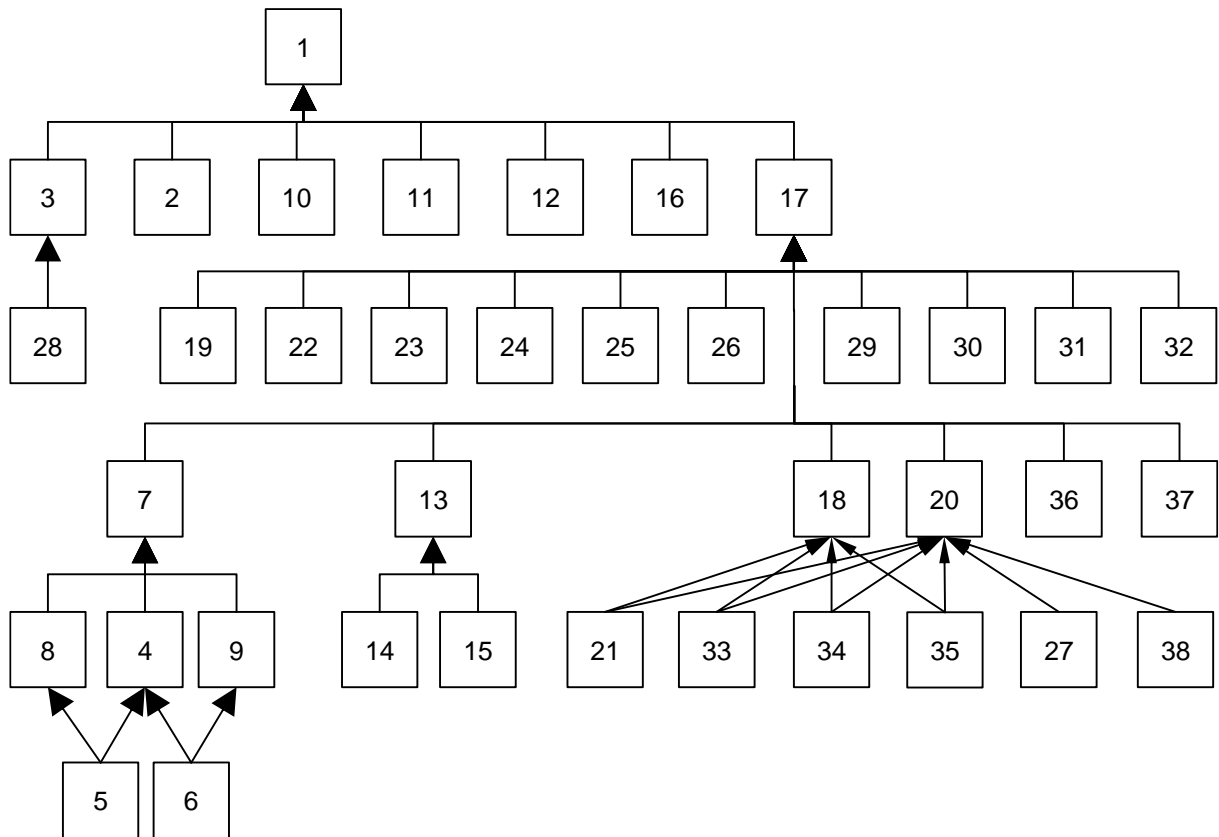
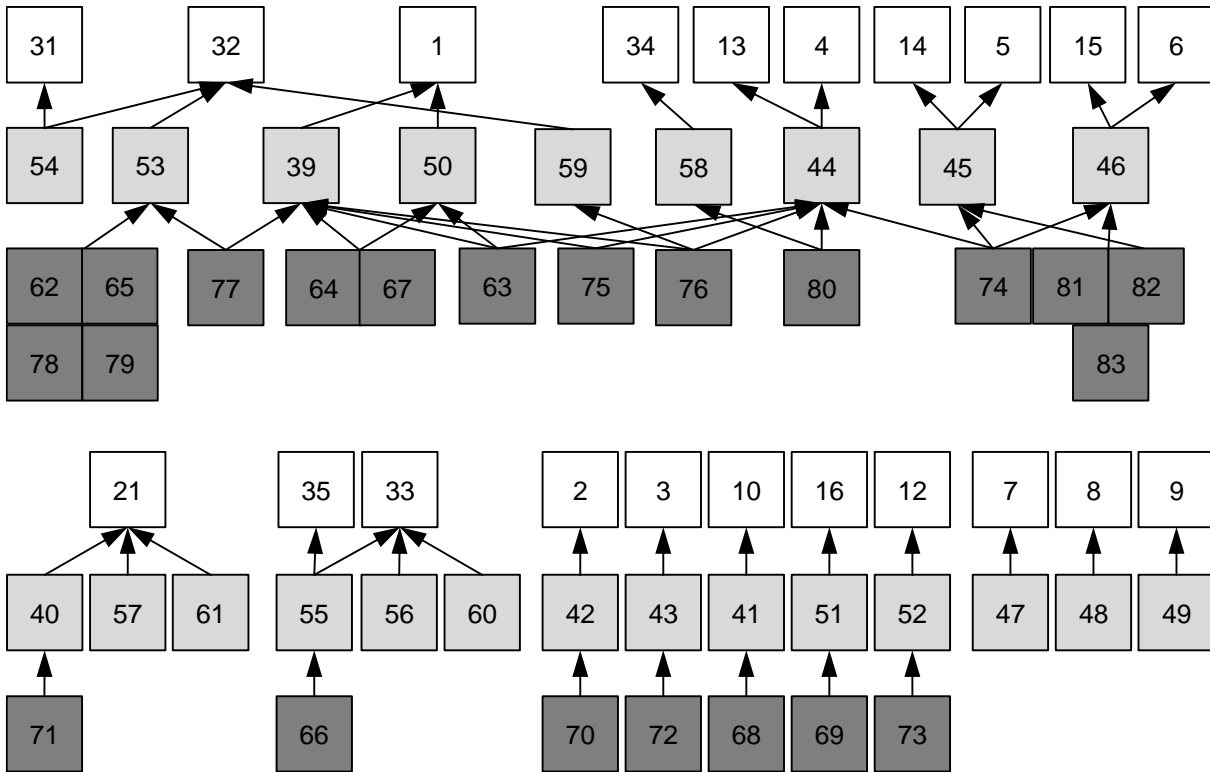


Figure 11-1 Inheritance relations among structural analyses



1: graph	14: active incidence set	27: articulation node
2: independent set	15: passive incidence set	28: isolated node
3: vertex cover	16: cut set	29: open sequence
4: adjacency set	17: connected component	30: closed sequence
5: active adjacency set	18: strong component	31: path
6: passive adjacency set	19: k-connected component	32: shortest path
7: reachable set	20: block	33: cycle
8: active reachable set	21: clique	34: triangle
9: passive reachable set	22: biclique	35: elementary cycles
10: separating set	23: start node	36: tree
11: feedback arc set	24: end node	37: spanning tree
12: edge cover	25: leaf node	38: bridge edge
13: incidence set	26: transit node	

Figure 11-2 Composition relations among structural analyses



1: graph	29: open sequence	57: no. of cliques per node
2: independent set	30: closed sequence	58: no. of triangles per node
3: vertex cover	31: path	59: no. of shortest paths per node
4: adjacency set	32: shortest path	60: no. of cycles per edge
5: active adjacency set	33: cycle	61: no. of cliques per edge
6: passive adjacency set	34: triangle	62: average path length
7: reachable set	35: elementary cycles	63: average degree
8: active reachable set	36: tree	64: relational density
9: passive reachable set	37: spanning tree	65: diameter
10: separating set	38: bridge edge	66: girth
11: feedback arc set	39: order	67: cyclomatic number
12: edge cover	40: order of clique	68: vertex connectivity
13: incidence set	41: order of separating set	69: edge connectivity
14: active incidence set	42: order of independent set	70: independence number
15: passive incidence set	43: order of vertex cover	71: clique number
16: cut set	44: degree	72: vertex covering number
17: connected component	45: active degree	73: edge covering number
18: strong component	46: passive degree	74: degree distribution
19: k-connected component	47: reachability	75: degree centrality
20: block	48: active reachability	76: betweenness centrality
21: clique	49: passive reachability	77: closeness centrality
22: biclique	50: size	78: snowball factor
23: start node	51: size of cut set	79: forerun factor
24: end node	52: size of edge cover	80: clustering coefficient
25: leaf node	53: distance	81: activity
26: transit node	54: path length	82: criticality
27: articulation node	55: cycle length	83: karatkevich number
28: isolated node	56: no. of cycles per node	

Figure 11-3 Derivation relations among structural analyses

11.5 Models from the literature review

Table 11-5 Models from the literature review (part 1 of 2)

ID	System	Model	Source	Reference	Nodes	Edges
P01	Spreader	G	WS	[Ameri et al. 2008]	19	31
P02	Sprinkler	G	WS	[Ameri et al. 2008]	21	29
P03	Timber structure	G	WI	[Björnfot & Stehn 2007]	14	15
P04	Tied rafter	G	WI	[Björnfot & Stehn 2007]	8	10
P05	Automatic gearbox	P	WI	[Bonjour & Micaelli 2010]	8	12
P06	Vacuum cleaner	C	WS	[Schmitz et al. 2011]	30	44
P07	Vacuum cleaner	C	WS	[Schmitz et al. 2011]	30	46
P08	Vacuum cleaner	C	WS	[Schmitz et al. 2011]	30	53
P09	Vacuum cleaner	C	WS	[Schmitz et al. 2011]	30	55
P10	Vacuum cleaner	C	WS	[Schmitz et al. 2011]	30	49
P11	Vacuum cleaner	C	WS	[Schmitz et al. 2011]	30	57
P12	Vacuum cleaner	C	WS	[Schmitz et al. 2011]	30	49
P13	Vacuum cleaner	C	WS	[Schmitz et al. 2011]	30	58
P14	Chain saw	C	WS	[Einögg 2009]	18	30
P15	Cell phone	Fu	WS	[Höltta-Otto & de Weck 2007]	11	16
P16	Desktop computer	Fu	WS	[Höltta-Otto & de Weck 2007]	23	38
P17	Desk phone	Fu	WS	[Höltta-Otto & de Weck 2007]	14	20
P18	Laptop computer	Fu	WS	[Höltta-Otto & de Weck 2007]	16	22
P19	Diesel engine	Fl	WI	[Keller et al. 2009]	21	24
P20	Diesel engine	G	WI	[Keller et al. 2009]	21	35
P21	Vacuum cleaner	Fu	NA	[Kreng & Lee 2004]	34	75
P22	Vacuum cleaner	C	NA	[Kreng & Lee 2004]	34	76
P23	Automobile	P	NA	[Lee et al. 2010]	17	37
P24	Diesel engine	P	NA	[Teseon 2011]	28	65
P25	Assembly cell	Fl	WS	[Strelkow 2010]	110	70
P26	Assembly cell	C	WS	[Strelkow 2010]	110	147
P27	Electric razor	P	NA	[Park et al. 2008]	9	12

Table 11-6 Models from the literature review (part 2 of 2)

ID	System	Model	Source	Reference	Nodes	Edges
P28	Combustion engine	Fl	NA	[Smaling & de Weck 2006]	32	45
P29	Combustion engine	Fl	NA	[Smaling & de Weck 2006]	32	20
P30	Combustion engine	Fl	NA	[Smaling & de Weck 2006]	32	23
P31	Combustion engine	C	NA	[Smaling & de Weck 2006]	32	60
P32	Automobile	G	WI	[Langer et al. 2010]	11	35
P33	Ball-pen	C	WS	[Maurer 2011]	8	11
P34	Aircraft engine	C	NA	[Maurer 2011]	7	10
P35	Ball-pen	C	WS	[Lindemann et al. 2009]	8	9

C: Contact; Fl: Flow; Fu: Function; G: Geometric; P: Product

NA: not available; WS: Work on system; WI: Workshop/Interview

11.6 Models from the design repository

Table 11-7 Models from the design repository (part 1 of 3)

System	Nodes	Edges	System	Nodes	Edges
air purifier	44	54	b and d screwdriver	46	59
digital alcohol detector	28	32	b and d sliceright	33	44
apple usb mouse	14	17	camera	35	52
asm volume 1	114	31	cassette player	34	23
asm volume 2	8	2	cd player	21	17
ball shooter	19	21	clorox readymop	19	26
bissell hand vac	39	42	colgate motion toothbrush	24	31
black 12 cup deluxe coffee	63	86	component basis	132	56
black 12 cup economy coffee	44	59	coolit drink cooler	15	17
black 4 cup regular coffee	38	54	cordless kettle	21	17
blowervac	11	15	cotton candy machine	36	27
bosch brad nailer	39	46	craftsman nextec multi tool	24	34
brake system	42	30	crest toothbrush	33	41
braun coffee grinder	14	18	dazey stripper	40	45
bugvac	26	30	delta circular saw	54	56
burton induction cooktop	18	27	delta drill	49	75
b and d can opener	28	56	delta flashlight	26	30
b and d circular saw attachment	33	38	delta jigsaw	40	27
b and d drill attachment	32	35	delta nail gun	64	72
b and d dustbuster	29	41	delta sander	37	33
b and d jigsaw	36	58	dewalt sander	35	50
b and d jigsaw attachment	40	48	digital scale	21	28
b and d mini router attachment	29	26	dirt devil vacuum	17	24
b and d palm sander	21	20	dishwasher	26	33
b and d power pack	45	55	dremel multi max	24	32
b and d rice cooker	59	64	dryer	29	28
b and d sander attachment	18	20	dual powered charging station	27	36

Table 11-8 Models from the design repository (part 2 of 3)

System	Nodes	Edges	System	Nodes	Edges
durabrand iron	41	49	kid smart smoke detector	16	17
dvd player	19	19	lawn mower	19	32
dyson air mutiplier	39	51	microfiber floor mop	11	14
e sky honeybee helicopter	63	52	milwaukee tubing cutter	58	69
electric stapler	51	71	milwaukee palm nailer	28	40
eyeglass cleaner	53	51	mini bumble ball	42	83
firestorm battery	19	38	mixer	33	61
firestorm circular saw	69	111	mr coffee iced tea maker	45	87
firestorm flashlight	28	42	neato vacuum cleaner	72	116
firestorm saber saw	56	62	oliso smart iron	64	90
firestorm screwdriver	28	2	oral b toothbrush	20	35
first alert basic smoke alarm	12	11	orion paintball gun	60	107
first shot nerf gun	34	44	polaroid pogo	38	40
game controller	29	49	power mat	19	18
garage door opener genie	43	79	presto salad shooter	25	44
ge microwave	47	90	proctor silex iron	36	63
giant bicycle	25	33	quickgrip irwin	13	20
grip right mini air nailer	13	20	random orbital sander	44	51
gse solar power module	13	16	razor scooter	53	84
hair trimmer	37	39	ridgid job max	22	38
health o meter digital scale	25	33	ridgid tube cutter	14	10
hitachi brad nailer	35	47	shopvac	33	61
holmes fan	37	33	skil circular saw	42	82
hot air popper	22	24	skil drill	57	94
iphone 3gs	25	42	skil flashlight	17	19
jar opener	73	1	skil jigsaw	47	78
juice extractor	30	23	slow cooker	24	25

Table 11-9 Models from the design repository (part 3 of 3)

System	Nodes	Edges	System	Nodes	Edges
snowcone maker	42	44	versapak sander	20	40
stapler	25	24	vibrating razor	14	16
stir chef	44	44	vise grip	13	25
supermax hair dryer	41	80	water pump	17	28
tippman paintball gun	55	70	westbend electric wok	31	61
toastmaster toaster	15	28	white 12 cup regular	40	83
tractor sprinkler	51	67	white 4 cup economy coffee	40	81
versapak circular saw	31	67	zip drive	23	36

11.7 Modelling and simulation details for the change case study

Table 11-10 Consistency relations leading to changes

Relation	Description
Adaptation	Both components must have the same size.
Not smaller than	The component must be at least as big as the other.
Not bigger than	The component must be at most as small as the other.
Bigger	The component must be bigger than the other.
Smaller	The component must be smaller than the other.
Only slightly bigger	The component may be only slightly bigger than the other.
Only slightly smaller	The component may be only slightly smaller than the other.
Space conflict	Both components use the same space. Therefore, the size increase of one component reduces the size of the other and vice versa.

Table 11-11 Priority rules for change requests

Rule	Description
All changes at once	Each change request is implemented immediately. If the requests exceed the resource limit the changes are chosen randomly.
Singulare changes first	First, the components without links to other components with pending change requests are changed. Then, the others.
FIFO – First in first out	The first change request is implemented first.
LIFO – Last in first out	The last change request is implemented first.

Table 11-12 Priority rules for change options

Rule	Description
Arithmetic mean	The requested options of all neighbors are averaged arithmetically.
Median	The median requested option of all neighbors is chosen.
Geometric mean	The requested options of all neighbors are averaged geometrically.
Harmonic mean	The requested options of all neighbors are averaged harmonically.
Hölder mean	The requested options of all neighbors are averaged using the second order Hölder mean.
Mode	The most often requested option of all neighbors is chosen. If two or more options are requested equally often the option is chosen randomly among them.
Mean (fixed, changed)	The requested options of all fixed and all changed neighbors are averaged arithmetically.
Mean (changed)	The requested options of all changed neighbors are averaged arithmetically.
First change	The requested option of the first change request is chosen.
Mode and first change	The most often requested option of all neighbors is chosen. If two or more options are requested equally often the option requested first is chosen.
Mode (changed)	The most often requested option of all neighbors is chosen. If two or more options are requested equally often the option is chosen randomly among them. Only the requests by changed components are considered.

11.8 Full results of the reconfiguration case study

Table 11-13 Average values of the metrics for each reconfiguration (changed and removed)

product change at component level (case A)			
metric	changed	unchanged	removed
degree	5.79	3.54	2.00
clustering coefficient	0.13	0.27	0.06
average distance	4.00	4.01	4.00
product change at module level (case B)			
metric	changed	unchanged	removed
degree	5.33	3.25	2.00
clustering coefficient	0.34	0.48	0.33
average distance	1.88	2.16	2.23
new handling strategy at module level (case C)			
metric	changed	unchanged	removed
degree	6.40	4.92	n/a
clustering coefficient	0.35	0.41	n/a
average distance	2.42	2.73	n/a
scale-up of product at module level (case D)			
metric	changed	unchanged	removed
degree	5.85	4.38	n/a
clustering coefficient	0.40	0.43	n/a
average distance	2.59	2.79	n/a

Table 11-14 Average values of the metrics for each reconfiguration (extended)

product change at component level (case A)		
metric	extended	unchanged
degree	4.88	3.55
clustering coefficient	0.08	0.27
average distance	3.92	4.02
product change at module level (case B)		
metric	extended	unchanged
degree	7.50	3.76
clustering coefficient	0.01	0.23
average distance	3.82	4.09
new handling strategy at module level (case C)		
metric	extended	unchanged
degree	5.17	3.00
clustering coefficient	0.27	0.48
average distance	1.92	2.16

Table 11-15 Level of significance of the observations for each reconfiguration (changed and removed; part 1 of 2)

product change at component level (case A)		
metric	changed vs. unchanged	level of significance
degree	>	17.90 %
clustering coefficient	<	3.05 %
average distance	<	89.42 %
metric	changed vs. removed	level of significance
degree	>	4.42 %
clustering coefficient	>	35.50 %
average distance	>	83.03 %
metric	unchanged vs. removed	level of significance
degree	>	1.39 %
clustering coefficient	>	0.11 %
average distance	>	47.73 %
product change at module level (case B)		
metric	changed vs. unchanged	level of significance
degree	>	18.33 %
clustering coefficient	<	35.93 %
average distance	<	12.31 %
metric	changed vs. removed	level of significance
degree	>	7.01 %
clustering coefficient	>	97.47 %
average distance	<	1.10 %
metric	unchanged vs. removed	level of significance
degree	>	11.60 %
clustering coefficient	>	69.01 %
average distance	<	66.00 %

Table 11-16 Level of significance of the observations for each reconfiguration (changed and removed; part 2 of 2)

new handling strategy at module level (case C)		
metric	changed vs. unchanged	level of significance
degree	>	29.54 %
clustering coefficient	>	53.27 %
average distance	<	3.89 %
scale-up of product at module level (case D)		
metric	changed vs. unchanged	level of significance
degree	>	2.19 %
clustering coefficient	<	66.13 %
average distance	<	4.87 %

Table 11-17 Level of significance of the observations for each reconfiguration (extended)

product change at component level (case A)		
metric	extended vs. unchanged	level of significance
degree	>	36.78 %
clustering coefficient	<	0.02 %
average distance	<	0.99 %
product change at module level (case B)		
metric	extended vs. unchanged	level of significance
degree	>	n/a
clustering coefficient	<	< 0.01 %
average distance	<	66.30 %
new handling strategy at module level (case C)		
metric	extended vs. unchanged	level of significance
degree	>	20.22 %
clustering coefficient	<	1.14 %
average distance	<	19.34 %

