DELIMITING STRUCTURAL AND DYNAMICAL SYSTEM ANALYSIS IN ENGINEERING MANAGEMENT

W. Biedermann, K. J. Diepold, U. Lindemann and B. Lohmann

Keywords: structural complexity management, hybrid dynamical systems, design structure matrix, complex system analysis

1. Structural and dynamical system analysis in engineering management

1.1 Challenges in engineering management

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. modularization), processes (e.g. agile development) and organizations (e.g. task forces). Models of the products, processes and organizations are fundamental for their design [Lindemann et al. 2009].

There are two major modelling approaches for 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 very precise analyses and deliver very specific results. They are usually used for detailed planning and optimization. The major drawback of dynamic-based models is the need for a lot of data 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 of general analyses and deliver rather fundamental results. 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 improve the management of engineering systems the strengths of both approaches are sought 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].

1.2 Procedure for integrated dynamical and structural analysis

[Diepold et al. 2010a] introduce the combination of structural complexity management and hybrid dynamical system modelling. Structural complexity management focuses on qualitatively describing the system elements and their connectivity, whereas hybrid dynamical system modelling puts the system's dynamic in focus. The major interface between both concepts is given by substructures of the system, which mainly determine its behaviour. Structural complexity management provides methods for identifying those substructures based on structural criteria such as cycles. They introduce a process to combine both methods and transform the results of structural analysis into a moderate mathematical model. The process is discussed in detail using a ball-pen as example system. Figure 1 shows the process and its major artefacts. However, they do not present any details on the models they transform into each other and do not present a set of substructures to identify system parts for detailed modelling.

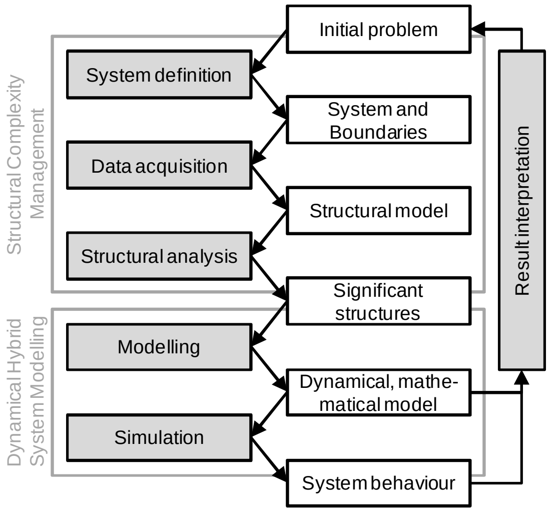

Figure 1. Modelling process for multi-dynamic mapping [Diepold et al. 2010a]

A modelling framework helping to transform a structural model into a hybrid dynamical onces is introduced in [Diepold et al. 2010b]. The framework consists of structural analysis, qualitative error estimation, dynamical modellingasformation, error estimation and data acquisition for refinement (if required). [Biedermann et al. 2010a] present a classification of approaches to refine both structure-based and dynamic-based models. [Diepold et al. 2010b] also describe the transformation from structure-based to dynamic-based models in detail. Finally, they present a new view onto the modelling process by [Diepold et al. 2010a]. They emphasize the iterative character of the process.. However, a remaining critical questions are which structural analyses are suitable for identifying refinement potential and for deducing dynamical behaviour.

1.3 Aims and research questions

The aim of this paper is to describe the limits of structure-based analyses and to specify to which degree dynamic analyses can be reasonably deduced from structure-based models. We work of three research questions:

1. Do structure-based models allow for deducing system behaviour?
2. Do structure-based models allow for deducing the entire system behaviour?
3. Do structure-based models allow for specifying the need for dynamic-based modelling?

2. Approach for comparing structural properties and simulation results

This paper compares the behaviour of system components with its structural properties. The case study uses component models of an assembly cell [Biedermann et al. 2010b]. The behaviour is determined via simulation. The simulation describes the system response when changing the size of a component. The simulation is summarized in the following, while 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 adpted 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, evaluating the simulation results.

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 dependences between them, and it is computaionlly handled as cellular automaton [Wolfram 1983]. In order to allow a simulation, each component is attributed with its volume as state variable. As a separately addressing of each component will led to a state explosion, a discrete set ("small", "medium", "large") is introduced and all components are matched to the set. 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 summarized in Table 1.

Table 1. Consistence relations

Relations between components	Explanation
at least, equal, at most	Element A must be at least, exactly or at most as big as elemt B
one size smaller, larger	Element A must be one size smaller or larger as elemt B.
at least, at most one size	Element A must be at least one size smaller or at most larger as elemt B.
conflictingt	The volumes of element A and B are conflicting, after a certain change

For each component 6468 simulations were run. The runs differ in change policy (for prioritizing and choosing among change options), maximum number of parallel changes, initial size and locking time (time delay) after changes. For instance, first in first out (FIFO), last in first out (LIFO) and changing all at once as prioritizing policy and arithmetic mean, median and mode as change options, to name just a few. The change option policy decides based on the incoming change commands of a node how its volume finally has to be adapted.

Each simulation is initialized 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, the number of size changes, the number of simulated time steps and the type of the final result were recorded. To compare the results with structural properties, five simulation metrics were computed for each component (representing an set of 6468 simulations):

- **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 a component's size. 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.
- **Standard deviation of the number of timesteps:** Each simulation covers several time steps until the states of components are not changed any more (steady-state). 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.
- **Number of new consistent solutions:** Each simulation results in one of three final states (according to the introduced set). If a new consistent solution is reached no consistency relation is violated and the size of at least on component differs from its original state. This is the intended outcome when changing components. If the changes of a component mostly result in new consistent solutions it can be handled easily under many boundary conditions. Thus, such components are prime candidates for introducing changes to a system's behaviour. This research examines if such components can be identified by structural analysis.
- **Number of fallbacks to the original solution:** If the sizes of the components do not differ from the original state at the end of the simulation, the system has fallen back to its original solution. This is not a desired outcome of an introduced change. If the changes of a component mostly result in fallbacks it must be handled individually and cannot be coverd by simple policies. Thus, changing such components should be avoided. This research examines if such components can be identified by structural analysis.
- **Number of inconsistent final states:** If at least one consistency relation is violated at the end of the simulation, but the system cannot be changed due to the selected change policies the simulation has reached an inconsistent final state. This is not a desired outcome of an introduced change. If the changes of a component mostly result in inconsistent states it must be handled individually and cannot be coverd by simple policies. Thus, changing such components should be avoided. This research examines if such components can be identified by structural analysis.

The components were structurally characterized by four metrics which were chosen based on [Cami and Deo 2008] and [Diepold et al. 2010]:

- **Degree:** The number of nodes to whom a node is connected to. This is one of the fundamental metrics describing a node and its role in the network. High values indicate hubs of a network which dominate the structure.
- **Clustering coefficient:** The ratio between the number of relations in a node's locality to the number of possible relations. High values indicate the node in embedded in a highly integrated, constraint environment.
- **Distance centrality:** The dictance is the length of the shortest path between two nodes. The distance centrality is a measure for the average distance of a node to all others. High values indicate central nodes.
- **Number of cycles per node:** A cycle is a closed sequence of relations connecting nodes. Other common terms are loop and circle. Each node can be part of several cycles. This metric gives the number of cycle each node is part of. Generally it is assumed that many cycles indicate very complex and unpredictable behaviour.

To examine the interrelation between structural and simulation metrics scatter plots were created. Figure 2 and Figure 3 show the obtained 20 plots. Due to data limitations we relinquished further statistical analyses such as analyses of means, correlations and significance.

3. Comparison of structural properties and simulation results

3.1 Observations concerning the number of changed components and timesteps

Figure 2 shows the scatter plots of the standard deviations and the structural metrics. We make seven major observations:

The plots for the number of changed nodes and the number of timesteps hardly differ. This can be seen by row-wisely comparing the right and left column of Figure 2. Thus, all findings for the number of changed nodes apply for the number of timesteps and vice versa. This observation was not expected as the simulation allows for parallel execution of changes. We assumed that the number of changed nodes and the number of timesteps are almost 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 2. Thus, components with high degrees show predictable, invariable behaviour. So far we did not analyse what behaviour these compoents exhibit in detail. We 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 at the moment as there are only two components with high degree (>12) in the case study at hand.

Components with high clustering coefficients show small deviations (<5). This can be seen in the second row of Figure 2. Thus, components with high clustering coefficients show predictable, invariable behaviour. So far we did not analyse what behaviour these compoents exhibit in detail. We 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. We assume that his observation is generalizable as there are 19 components with high clustering coefficient (>0.5) in the case study.

Components with high or low distance centrality show small deviations (<5). This can be seen in the third row of Figure 2. Thus, components with high or low distance centrality show predictable, invariable behaviour. So far we did not analyse and have no conjecture what behaviour these compoents exhibit. However, this observation is hardly generalizable yet as there are only one component with low distance centrality (<15) and only two components with high distance centrality (>40) in the case study at hand.

The number of cycles does not correlate with high (>14 for number of changed nodes and >12 for number of timesteps) or low (<5 for both cases) deviations. This can be seen in the fourth row of Figure 2. 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. We assume that this observation is generalizable as there are several components for each value range.

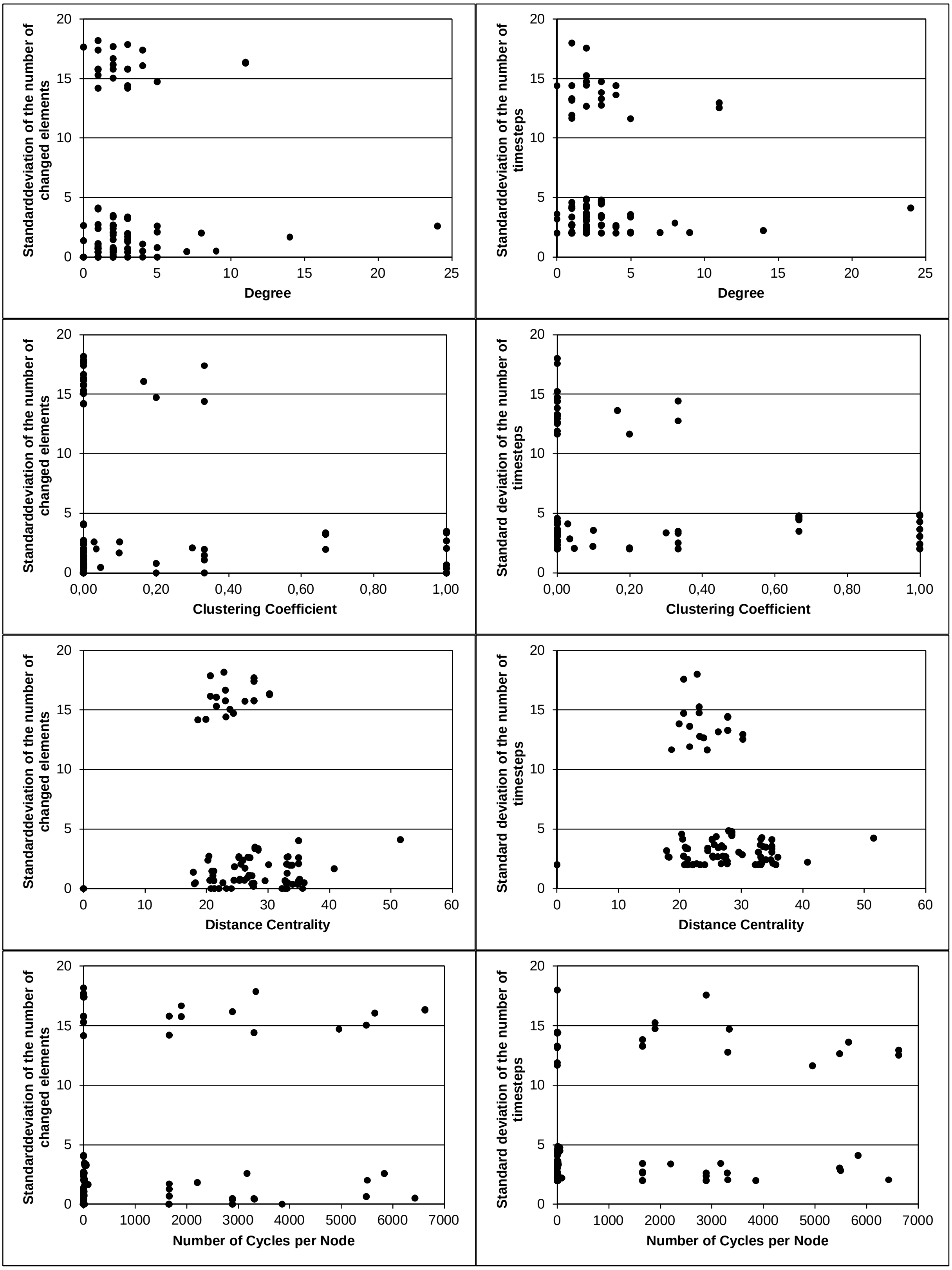

Figure 2. Scatter plots of the standard deviations of the number of changed elements (left column) and of the number timesteps (right column) versus the degree (first row), the clustering coefficient (second row), the distance centrality (third row) and the number of cycles per node (fourth row)

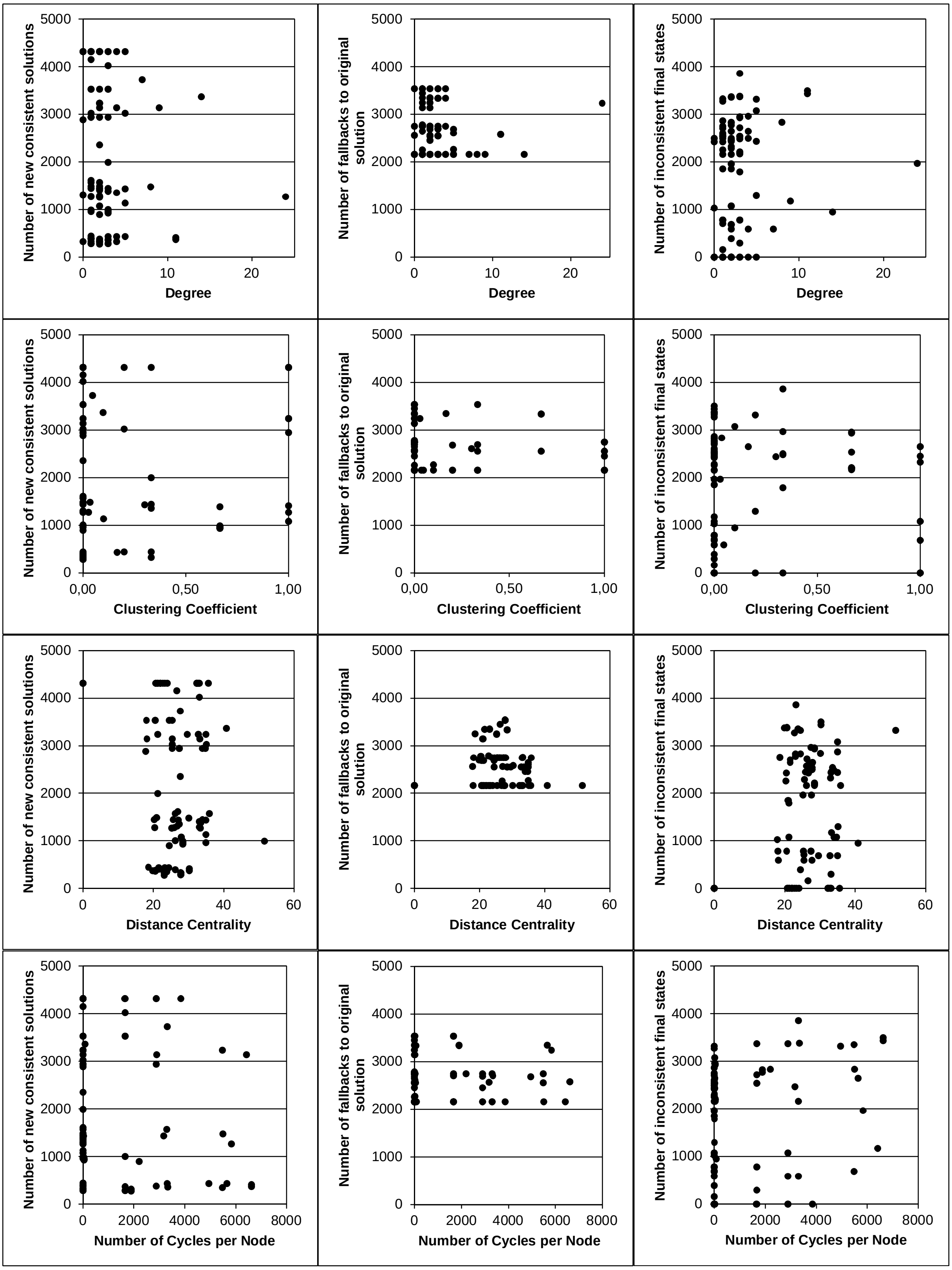

Figure 3. Scatter plots of the number of new consistent solutions (left column), of fallbacks to the original solution (middle column) and of inconsistent final states (right column) versus the degree (first row), the clustering coefficient (second row), the distance centrality (third row) and the number of cycles per node (fourth row)

There are two groups of components: one group with high deviations (>14 for number of changed nodes and >12 for number of timesteps) and one group with low deviations (<5 for both cases). This can be seen in all plots in Figure 2. Thus, there are two groups of components: one group with predictable, invariable behaviour and one group with unpredictable, variable behaviour. As there is a quite big gap in between the groups, we assume that there is a fundamental characteristic that differentiates the components of the groups. An efficient analysis which allows for identying the groups before modelling, simulating and analysing the behaviour, will greatly enhance the effiency of system analysis. The low variety group should be modelled using simple models, which do not require a lot of data. Whereas the high variety group should be modelled using complex models.

The groups do not correlate to one of the four metrics. This can be seen in any plot in Figure 2 as there is no clear separation of the groups on the x-coordinate. Thus, none of the researched metrics allows for separating the components into groups according to the predictability of the behaviour. Yet, we 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 the consistency relations. However, we did not perform an in depth analysis on that, yet.

3.2 Observations concerning the resulting solutions

Figure 3 shows the scatter plots of the solution numbers and the structural metrics. We make four major observations:

Clustering coefficient and number of cycles per node show no correlations to the number of solutions. This can be seen in the second and fourth row of Figure 3. Thus, both metrics do not allow for identying nodes for particularly easy or complicated changes. We assume that this observation is generalizable as there are several components for each value range.

Components with high degree show high numbers of fallbacks and slightly below average numbers of new solutions and inconsistent final states. This can be seen in the first row of Figure 3. Thus, changing components with high degree should be avoided. We conjecture that highly connected nodes are too constraint to be changed sustainably. However, this observation is hardly generalizable yet as there are only two components with high degree (>12) in the case study at hand.

Components with low distance centrality show very high numbers of new solutions and small numbers of fallbacks and inconsistent final states. This can be seen in the third row of Figure 3. Thus, such components are prime candidates for introducing changes to a system. We have no conjecture to explaine the observation. However, this observation is hardly generalizable yet as there is only one component with low distance centrality (<15) in the case study at hand.

Components with high distance centrality show low numbers of new solutions and high numbers of fallbacks and inconsistent final states. This can be seen in the third row of Figure 3. Thus, changing such components should be avoided. We have no conjecture to explaine the observation. However, this observation is hardly generalizable at the moment as there are only two components with high distance centrality (>40) in the case study at hand.

4. Discussion, conclusion and future work

The results suggest that highly-connected (indicated by the degree) and highly-embedded (indicated by the clustering coefficient) components have very predictable behaviour. One explanation is that such highly-connected components can impose changes onto their locality. Likewise highly-embedded components are too constraint to show highly variable behaviour. The observations concerning the distance centrality can be explained analogously. We suggest to use degree and clustering coefficient for choosing appropriate dynamic models for the components. As high values indicate predictability these components should be modelled rather simple.

It is generally assumed that many cycles lead to variable and unpredictable behaviour. However, we cannot confirm this assumption concerning the case study at hand. We have no explanation for our observation. However, it might be stated that number of cycles can be seen as weighting factor (boost) in addition to degree and clustering coefficient when choosing appropriate dynamic models for the components. In other words, if a compontes, which should be modeld rather complex based on degree and clustering coefficient, appreas in many cycles the need for a complex model is reinforced.

There are two groups of components: one group with high deviations and one group with low deviations. This indicates that there is a hidden variable determining the variability of the behaviour which has to be identified in future research.

Only the observations and conjectures concerning the clustering coefficient are generalizable. There are 19 components with high (> 0.5) clustering coefficients. The other observations and conjectures are at best hints which have to be validated and support by future research.

Our results show that behaviour can be deduced from structural models. However, the behaviour of all components of a system cannot be predicted. Thus, dynamic models cannot be replaced by purely structural models. Based on our observations and considerations concerning generalizability, we suggest to use the clustering coefficient for choosing appropriate dynamic models. High values indicate predictable behaviour which can be described in simple models.

Our results show that some components show highly variable behaviour. The most pressing task is to identify these components a priori based on structural models. In doing so, the modelling efforts for dynamic analysis can be reduced drastically. One secondary task is testing our conjectures in other systems and domains.

Acknowledgements

This research was made possible through the generous funding by the German Research Foundation (DFG) in the context of project A2 (“Modeling and analysis of discipline-spanning structural criteria and their impacts on product development processes”) and the project A3 (“Tool-based methods for complexity management: stability and robustness analyses”) within the Collaborative Research Centre SFB 768 (“Managing cycles in innovation processes”).

References

Biedermann, W., Diepold, K. J., Lindemann, U., “Modellabstufungen zur Betrachtung struktureller und dynamischer Komplexität.”, Tag des Systems Engineering, Maurer, M., Schulze, S.-O. (Eds.), Hanser München, 2010, pp. 97-106.

Biedermann, W., Diepold, K.J., Lindemann, U., “Entscheidungsstrategien zu Festlegung und Priorisierung von Änderungen”, Tag des Systems Engineering, Maurer, M., Schulze, S.-O. (Eds.), Hanser München, 2011, pp. 141-150.

Biedermann, W., Strelkow, B., Karl, F., Lindemann, U., Zaeh, M., “Reducing Data Acquisition Effort by Hierarchical System Modelling.” 12th International DSM Conference, Cambridge 2010.

Browning, T. R., “Applying the design structure matrix to system decomposition and integration problems: A review and new direction.”, IEEE Transaction on Engineering Management, 48, 3, 2001, pp. 292–306.

Cami, A., Deo, N., “Techniques for Analyzing Dynamic Random Graph Models of Web-Like Networks: An Overview.”, Networks, 51, 4, 2008, pp. 211–255.

Diepold, K. J., Biedermann, W., Eben, K. G. M., Kortler, S., Lohmann, B., Lindemann, U., “Combining Structural Complexity Management and Hybrid Dynamical System Modelling”, Proceedings of the 11th International Design Conference - DESIGN 2010, Marjanović D., Štorga M., Pavković N., Bojčetić N. (Eds.), Dubrovnik, Croatia, 2010, pp. 1045-1054.

Diepold, K. J., Winkler, F. J., Lohmann, B., „Systematical Hybrid State Modelling of Complex Dynaamical Systems: The Quad-I/HS Framework.“ Journal of Mathematical and Computer Modelling of Dynamical Systems, 16, 4, 2010, pp.347-371.

Lindemann, U., Maurer, M., Braun, T., “Structural Complexity Management – An Approach for the Field of Product Design.”, Springer Berlin, 2009.

Strogatz, S. H., “Exploring complex networks.” Nature, 410, 6825, 2001, pp. 268-276.

Wolfram, S., “Statistical mechanics of cellular automata”. Reviews of Modern Physics, 55, 3, 1983, pp. 601–644.

Wieland Biedermann
Scientific assistant
Technische Universität München, Institute of Product Development
Boltzmannstr. 15, 85748 Garching, Germany
Telephone: +49 89 289-15129
Telefax: + 49 89 289-15144
Email: biedermann@pe.mw.tum.de
URL: http://www.pe.mw.tum.de