INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

PROCESS ANALYSIS USING STRUCTURAL METRICS: A COMPREHENSIVE CASE STUDY

M. Kreimeyer, N. Bradford and U. Lindemann

Keywords: structural complexity, process, analysis, metric

1. Introduction

Efficient process management is an important success factor for any design process nowadays, with products (and hence processes) becoming more and more complex [Lindemann, et al., 2009]. As a part thereof, the analysis of process models has been in focus of various research activities for many years now, striving to support the goal-oriented improvement of such processes with different strategies.
This paper demonstrates the use of complexity metrics to this end, based on a large case study. Analyzing complex processes
Process analysis is made difficult especially by the need to describe a process without deep knowledge of the process itself. Often, the modeling process is so complex that it is impossible to extract the actual knowledge from the model, especially when it contains a multitude of entities and relations. Often, however, the analysis of a process is, in fact, the analysis of a process model that has been generated in a previous step or that already existed (at least in part).
In order to bridge this gap, different means of analysis can be used, e.g. simulation approaches, the comparison of different process scenarios, regarding the structure of the process [Lindemann et al. 2009], or others. In all cases, the goal is to generate indications and inferences about the process behavior and possible improvement measures. Here, the focus is put of structural metrics that only have come up recently and have, therefore, seen little attention yet; however, they demand much less effort in their application while allowing a systematic high-level analysis to extract potential weak spots in a process that merit attention [Kreimeyer, et al., 2008].

1.1 Research methodology and structure of this paper

So far, only few studies using structural analysis are available [e.g. Braha and Bar-Yam, 2007, Kreimeyer 2008, Schlick 2008]. These studies only represent partial views at each time; a comprehensive overview providing explicit results was not achieved, due to the specific scope of each research. Within this paper a wider overview is given, to provide a complete overview of the current applicability of such structural metrics. Therefore, an existing, published process model was chosen, in order to be able to compare newly calculated and existing results. To do so, section 2 contains information about structural complexity management using metrics, and section 3 introduces the process in focus. In section 4, the analysis procedure and its results are presented. Chapter 5 concludes the paper by providing implications for research and industry as well as directions for future reseach.

2. Metrics-based management of structural complexity

So far, the use of metrics to formalize an analysis is little regarded in engineering design [Bashir, 1999], especially when it comes to assessing the structure of a process, i.e. the constellation of how the entities of a process (tasks, documents, milestones,…) interact [Mendling, 2008]. One approach towards this goal is the use of structural metrics [Kreimeyer, et al., 2008].

2.1 Complexity of engineering design processes

Complexity is present in many disciplines. Commonly, complexity means consisting of parts or entities not simply coordinated, but some of them involved in various degrees of subordination; complicated, involved, intricate; not easily analyzed or disentangled. In engineering, complexity generally addresses the coupling of the entities of a technical system [Lindemann, et al., 2009], and software science focuses on assessing program code for its complexity, and thereby the risk of introducing errors into the code. Of course, many other definitions for specific disciplines prevail, too.

2.2 Structural complexity management

Structural complexity management [Lindemann et al. 2009] is often seen as having evolved out of engineering projects that were accompanied by the paradigm of Systems Engineering. While structural complexity generally regards technical (i.e., planned) systems, in parallel, Network Science [Newman 2002] describes complex systems of random or natural origin, e.g. social networks.

In this context, IEEE defines process complexity as "the degree to which a process is difficult to analyze, understand or explain. It may be characterized by the number and intricacy of activity interfaces, transitions, conditional and parallel branches, the existence of loops, roles, activity categories, the types of data structures, and other process characteristics" [Cardoso, 2006].

In process management, structure therefore refers to patterns of entities and relations of a process. In particular, workflow patterns represent the basic decision structures of a process as possible constellations of splitting and joining the control-flow of a process. [Cardoso, 2006]. Similarly, such patterns can be used to detect possible errors in a process, as [Mendling ,2008] proposes.

Extending this approach, the structure can be also be regarded to understand the behavior of the system, and both structural complexity management and network science propose: From a structural point of view, a system can be disentangled into a network-like model of entities and their relations. These entities can be of different classes, e.g., documents, organizational units, and work packages. Each kind of entity represents a specific view, called a domain. The purpose of a domain is to create homogeneous networks that allow elements to be compared during analysis [Lindemann et al. 2009]. The term domain can, therefore, be defined as a specific view of a complex system, comprising one type of entity. Each domain is, typically, accompanied by a specific relationship type that is defined as a class of relations (e.g. domain A "generates" entities of domains B).

2.3 Complexity metrics for process analysis

Metrics are a means of representing a quantitative or qualitative measurable aspect of an issue in a condensed form., and structural metrics have seen much attention especially in software engineering and network science. Many approaches from these disciplines can be transferred and adapted to fit structural complexity management in engineering design. Espcially software metrics are highly relevant to process management, as a software program and the control-flow graph of a process are very similar; several authors have drawn attention to the fact that executing a software is much like running a workflow or a process [Cardoso, 2006].

Typically, metrics are used for three different purposes in engineering design: Estimation, monitoring, and performance measurement. Yet, there is generally little specific work on metrics for engineering design processes available. This is mainly because product development has the nature of a mental exercise and because of a lack of easily identifiable items to measure [Bashir, 1999]. It is true that the existing metrics, therefore, remain either highly specialized, or they are conceptual and hard to apply.

Commonly, metrics are not independent of each other but can be organized in a measurement system (according to, e.g., focus, goal, granularity). This enables the systematic and goal-oriented employment of metrics. This is especially important in regards to the structural analysis of a process, as a metric can only be purposeful in the context of a goal and the related semantics; metrics, therefore, cannot be designed without a meta-model that provides a semantic context to later interpret

the metric. This meta-model needs to provide the domains and relationship types that are used to model a process and that help give significance to the results of a measurement.
A strategy of the use of structural metrics is to identify structural outliers, i.e., such entities of a process that significantly stand out from the rest of the system with regard to a specific pattern. Of course, statistical significance cannot be reached for the analysis of most process models, as common process models only have a limited number of entities, and, therefore, the population of the analysis will be, from a statistical point of view, too limited to obtain a mathematically sound significance level or p-value. Rather, a structural outlier can be identified using the Pareto principle. The structural significance each metric provides can be used to investigate the nature of the outliers to guide further improvement measures. Of course, relevant outliers need to be be logical in the overall context of a system. While e.g. the first task of a process will most likely stand out concerning its ability to reach all other tasks in the process, it is not a relevant outlier, mostly.
The structural metrics in this paper are taken from the set of metrics shown in [Kreimeyer, 2010]. They have been developed using the above principles and represent a collection that was compiled out of different disciplines. The overall set of structural metrics consists of 52 metrics, of which, however, some cannot be computed yet and others only apply to a model of two or more domains. Therefore, in the following, only those 32 metrics that can be applied to a process model of one domain as described in the following section are used. All metrics are presented in section four.

3. The process in focus

The analyzed development process (available at http://necsi.org/projects/braha/largescaleengineering) represents 26 weeks of General Motors' automotive development separated into phases: Expert opinion phase, quick study phase and integrated vehicle concept model and o.d. deliverables phase. The process is described and analyzed in [Braha and Bar-Yam, 2007].

3.1 The process and the process model

Within the process, 120 tasks are linked in a Design Structure Matrix (DSM) containing 417 immediate directed relations. The tasks are accomplished by 19 organizational units. Each set of tasks belonging to an organizational unit is referred to as a module.

Table 1. Most important tasks in the process [Braha & Bar-Yam 2007]

T1	Develop Nine Box Summary	T88	Run Updated Workload Model
T2	Provide Preliminary 2-D Sketches of Vehicle	T89	Update Financial Assessment and Finalize Business Case
T3	Identify Target Architectures	T90	Update Decoupled Development Plan
T6	Provide Key Volume Drivers Chart	T91	Review Quick Study Deliverables
T10	Develop Critical Product Characteristics / Key Voices	T98	Finalize Body BOM
T11	Set Engineering Target Parameters (Concept Technical Descriptors)	T111	Create Physical/Virtual Models
T33	Run Initial Workload Model	T112	Assess Risks in Performance Requirements
T37	Recommend Final Architecture	T114	Develop Manufacturing Program Timing Plan
T72	Track Total Vehicle Issues	T115	Provide Final Volume Forecast
T73	Maintain Vehicle Mainstream Chart and Update Engineering Product Content Sheet	T116	Develop Final Integrated Concept Vehicle Model
		T117	Develop Option Plan for NASB Review
T83	Generate Surface Limits for Bodyside	T118	Review Integrated Vehicle Concept Model and O.D. Deliverables
T85	Conduct Marketing Clinics	T119	Provide Transition to Vehicle Program After Option
T86	Provide Customer's Perspective to Option Team	T120	Provide Transition to Program Quality Manager

The model was build from interviews with engineers and from design documentation. For each task, it was asked 'Where do the inputs for the task come from?' and 'Where do the outputs generated by the task go to?'. The answers were used to construct the network of information flows. In the following, the tasks shown in table 1 will occur repeatedly and are therefore listed hereas T1 to T120.

3.2 Existing results towards the analysis

In an earlier analysis of the process [Braha and Bar-Yam, 2007] the following main results were elicited. They concern especially the small-world properties (i.e. most nodes are not adjacent but reachable via a short average path length) and the degree-related properties (i.e. direct coupling among immediate neighbors) of the involved tasks:

Braha and Bar-Yam consider the process to exhibit clear small-world properties. Accordingly, the task network's entities have a relatively high cluster coefficient, whereas the characteristic path length is relatively short and approximately equal to a characteristic path length of a random graph having the same number of nodes and edges. A modular organization (defined by a higher degree of internal information exchange than across the borders of modules) was found to be a consequence of high cluster-coefficients and small word properties.
They furthermore identify an imbalance concerning the relation of out-degrees and in-degrees (also called activity and passivity). Most tasks were found to have relatively low in- and out-degrees, whereas few have high degrees. Those few having a high out-degree, or respectively passive (high in-degree) tasks are characterized as information generators (or information consumers, respectively). In turn, tasks with a high in-degree have a low out-degree and vice versa. The process is dominated by a small number of such tasks with either a high in- or out-degree.
Braha and Bar-Yam declare their results as typical for product development processes, with the following consequences: The most effective way of improving the overall process is to improve the central, dominating tasks, similar to the concept of structural outliers. Secondly, they conclude that a failure of those tasks is likely to impede the correct function of the overall process.

4. Process analysis using structural complexity metrics

New results were elicited during a new and more comprehensive analysis of the vehicle development process, using a set of structural metrics provided by [Kreimeyer, 2009]. Table 2 lists the 34 structural metrics used here. For each, the relevant dataset is listed in order to generate meaningful results. The metrics are arranged by categories related to the underlying structural patterns.

Table 2. Overview over the applied structural metrics (from [Kreimeyer 2009])

	Overall process	Tasks	Modules
Size and density			
Number of domains	x		
Number of nodes	x		
Number of edges	x		
Number of classes	x		
Number of interfaces between domains	x		
Number of edges per node	x		
Relational density	x		
Number of unconnected nodes	x		
Adjacency			
Activity / Passivity		x	
Degree correlation (nodes)		x	
Degree correlation (edges)		x	
Degree distribution		x	
Fan criticality			x
Synchronization points / distribution points		x	
Number of independent sets	x		
Cycles			
Number of feedbacks	x		
Number of cycles per node		x	
Number of clycles per edge		x	

	Overall process	Tasks	Modules
Attainability			
Reachability of a node		x	
Closeness		x	
Proximity		x	
Relative centrality (based on between-ness)		x	
Paths			
Number of paths	x		
Path length	x		
Hierarchies			
Height of hierarchy	x		
Width of hierarchy	x		
Snowball-factor		x	
Forerun-factor		x	
Tree-robustness			x
Clustering			
Number of cliques	x		
Cluster-coefficient (local)		x	
Cluster-coefficient (global)		x	x
Module quality 1 (flow of information)			x
Module quality 2 (compactness)			x

4.1 Analysis results

The results of the metrics generate distributions, within which the individual values can be compared to identify relevant outliers. Three views can be identified that each related to a distinct dataset. The overall process model as a whole, the tasks individually, and the modules as formed by the tasks belonging to the different organizational units. Thus, the results are organized accordingly.

4.1.1 Metrics for the overall process model

At first, a basic analysis of processes is the calculation of metrics concerning size and density as well as metrics delivering characteristic values for overall networks. The number of domains in the MDM

is two: Tasks and organizational units. The number of nodes is 139: 120 tasks and 19 organizational units. The number of edges is 537, of which 417 edges are entries of the task-task DSM. The number of classes (i.e. number of different kinds of nodes) is 139, equally. Accordingly no node is regarded repeatedly within the process model, which sometimes happens in process models when one task is instantiated several times. The number of interfaces between domains is 120, i.e. each task is executed by exactly one organizational unit. The number of edges per node is 3.457 for the task-task netowrk and 3.891 for the overall network (including org. units). Respectively, the relational density is 0.029 and 0.028. Both values show that a rather low part of all possible connections is exhausted and the process is rather linear. This concurs with the initial process model that can be triangularized easily, i.e. the task sequence can be put into an ideal order without severe conflicts. The number of unconnected nodes, which could reveal possible mistakes in the process model, is zero. The number of independent sets (i.e. the number of sets of tasks accomplished concurrently and independently from each other, as found when banding the respective DSM) is eight, i.e. the process can be broken down into eight phases. The number of paths across the overall process is especially useful for estimating the importance of root nodes. From root node T1 36 paths lead to the process' five end nodes, whereas 33 paths lead from root node T2, leaving both starting tasks relatively equal in their impact. The average path length of these shortest paths is 3.6 between T1 and the end nodes and 5.6 between T2 and the end nodes, showing that information spreads faster throughout the process starting at root node T1. This metric, although it describes connection between tasks, concerns the overall process as properties of start and end nodes. Figure 1 shows the number of paths and the according lengths in brackets.
Figure 2 vizualizes the different paths as a graph: The maximum height of the process' hierarchy, i.e. the number of levels from start to end nodes, is 4. The width of the process' levels (the number of nodes per level) is 2 on first level (i.e. the start nodes), 28 on second level, 50 on third level, 35 on fourth level and 5 on fifth level (i.e. the end nodes).

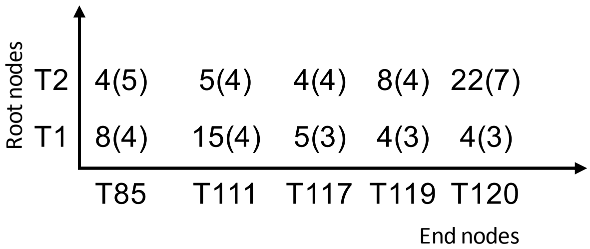


Figure 1. Number (length) of paths

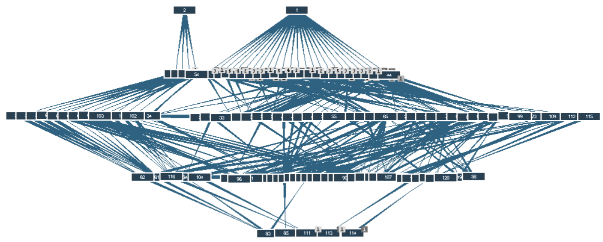


Figure 2. The five levels of the process

The process has two start nodes, namely T1 and T2, and five end nodes: T85, T111, T117, T119 and T120. The maximum nesting depth, i.e. the number of splits retraceable to a root node, is 100 for root node T1 and 33 for root node T2, showing again a higher influence of root node T1, as the process bifurcates noticeably more from this task. The number of cliques, i.e. the number of complete clusters within the network, is zero, i.e. no groups of tasks that are completely mutually connected exist within the model. The global cluster-coefficient (quotient of the sum of all cluster-coefficients per node and the number of information distributors) is 0.27, indicating that many tasks are likely to be coupled more intensely than the number of cliques shows; this potential for coupling relies on the concept that two tasks connected to a third task are likely to be interrelated because they are coupled to a third task in the same way. The number of feedbacks within the process, i.e. the number tears in a triangularized DSM, is 24, a rather low percentage (5.75% of all 417 connections), showing that the overall iterative nature of the process is broken down rather well into only few intended relations.

4.1.2 Metrics per task

The majority of structural metrics is applied to compare the entities of just one domain, i.e. the tasks of the task-task-DSM. As shown in Table 2, adjacency and attainability are the categories concerning most metrics applicable on the behavior of the correlations between tasks. Metrics referring to adjacency and attainability are predestinated for measuring the importance of single entities for the function of the complete network [Kreimeyer 2009].

The most basic metrics are activity and passivity (also referred to as out- and in-degree) of a node. Figure 3 shows a concurrent plot of both metrics per node. Four tasks (T72, T73, T65 and T37) stand out most, accordingly being highly active and highly passive at the same time. The four fields inert, active, passive and critical are defined by the average values for both axes. The start nodes T1 and T2 are positioned on the activity-axis with a value of 0 for passivity, as they only deliver information. However, T1's out-degree is nearly five times higher than the value for T2, indicating a higher initial impact of T1 onto the overall network.

The degree correlation can be based on edges as well as on nodes. The representation of the correlation based on nodes as in figure 4 reveals a high number of connections between nodes with values of one or two for activity or passivity. Accordingly, most nodes within the process have relatively low in- and out-degrees at the same time; at the same time, the correlation plot shows that many nodes are connected with similar in- and out-degrees, as the diagonal axis of the plot contains many non-zero entries. Similarly, the representation of the correlation based on edges (not shown) indicates that 64% of the edges link two nodes both having more than one incident as well as more than one outgoing edge, i.e. most information transfers between two tasks will be based on several inputs into the first task and generate more than one output at the second task. Theconclusion of both correlations is that a major part of the network consists of connections between nodes with low degrees, of which most have a higher degree than one (i.e. each task being coupled to more than one other task). Nonetheless, there are twelve highly important edges that are the only connection between the nodes (i.e. directed forwarding between two tasks).

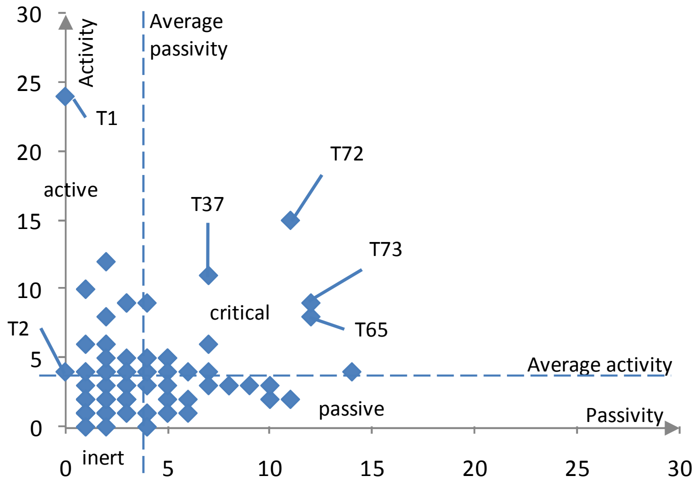


Figure 3. Activity and passivity per task

passive degree

active degree	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	2	1	0	2	0	0	0	0	0	0	0	0	0	0
1	0	12	6	2	2	2	1	0	0	0	0	0	0	0	0
2	0	5	11	7	0	1	1	0	0	0	1	1	0	0	0
3	0	1	2	4	3	1	0	1	4	1	1	0	0	0	0
4	1	2	6	4	4	1	1	1	0	0	0	0	0	0	1
5	0	0	1	1	3	4	0	0	0	0	0	0	0	0	0
6	0	1	2	0	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
9	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0
10	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
12	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4. Degree correlation (nodes)

The degree distribution reveals the occurrence of similar in- and out-degrees within the process. The plot underlines the occurrence of low degrees in the process, pointing to a hub-and-spoke-like structure of the overall process (i.e. the network is a scale-free network, like many typical collaboration structures [Newman 2002]). High degrees appear seldom, as figure 5 shows.

The representation of the synchronization and distribution points produces no new results as all entries within the DSM have the value one. Accordingly the representation is identical to the activity/passivity plots. In other cases, if e.g. a weighted DSM is used, these metrics would be able to underline the importance of a task not just based on the degree but also on its coupling strength.

The active and passive reachability describe the number of nodes a designated node is able to reach (or the number of nodes that can reach this designated node, respectively). In a process analysis, these metrics are very important, as they show the spread of information (and errors) across the overall process, thus estimating the impact (and impactedness) thereof for each task. They therefore extend the picture generated by the degree across not just adjacent tasks but across all tasks. The plot of both metrics per node (figure 6) a few highly actively and passively reachable tasks: T118, T112, T89, T90, T91and T75. These tasks are therefore highly integrated into the flow of information through the process and play an important role in the supply of information of all other tasks.

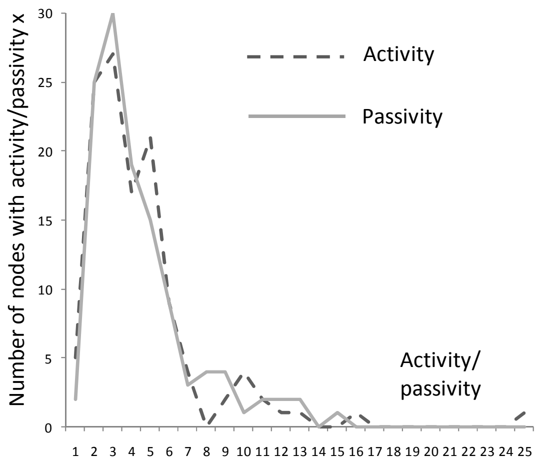


Figure 5. Degree distribution

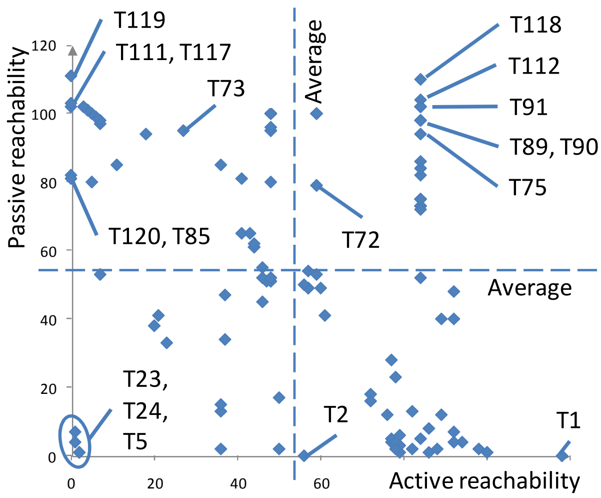


Figure 6. Active and passive reachability

The active and passive proximity are calculated by summating the rows (columns, respectively) of the distance matrix (listing the shortest path between any pair of tasks, zero if not reachable), i.e. describing the distance of one task to all others. As outliers for the active and passive proximity the following tasks appear: T75, T83, T89, T90, T98, T114, T155 and T116, (figure 7). Once again, tasks T75, T89 and T90 (cf. reachability) seem to be of higher importance for the function of the overall network, a relatively high average path length represents high impact, as a high number of nodes positioned on according paths are involved into the incident and outgoing flow of information.

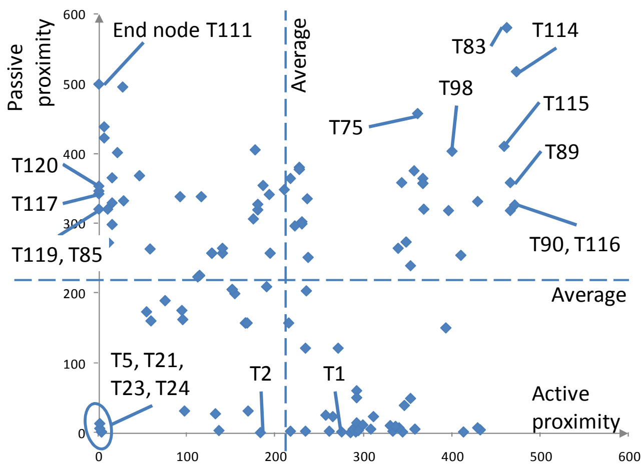


Figure 7. Active and passive proximity

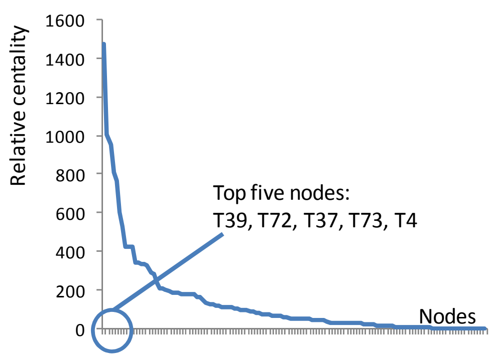
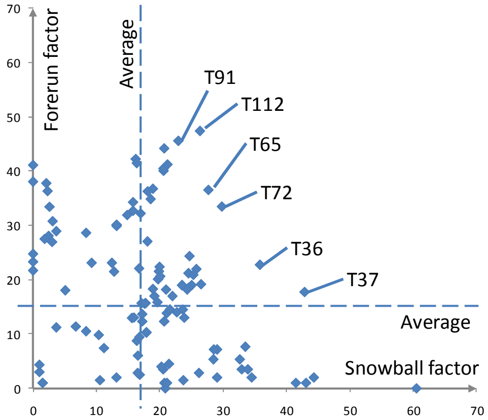


Figure 8. Relative centrality

In a comparable manner, the relative centrality counts the number of shortest paths between any two nodes that cross a designated node: The higher the value, the more information flows go via a designated node. The following tasks stand out: T39, T39, T4, T72 and T73 (figure 8).

Another pair of meaningful metrics for the estimation of influence of entities are snowball factor and the forerun factor. They assess the outgoing (incoming, respectively) hierarchy of reachable nodes with decreasing impact for nodes that are farther away: They calculate as the sum of the products of width and height of the level in the hierarchy, weighted by to the inverse of the shortest path length to the root node. They thereby relativize the active and passive reachability, as nodes that can be reached but that are far away and have little impact are not counted as importantly. Figure 9 shows the plot for both metrics per node. Here, tasks T36, T37, T65, T72, T91 and T112 show up, having high values for both metrics. The distribution of the values (figure 10) shows that, each time, only a few nodes have high influence onto the process. Those particular nodes are one start node, T1, as well as the tasks T3, T10, T11 and T37. Start node T2 only has the 44th position in this ranking, which underlines the much higher importance of root node T1. The plot of all values for forerun factors, however, shows a more linear distribution, indicating that few tasks dominate the spread of information, while the tasks rely more homogeneously on the information intake from other tasks.



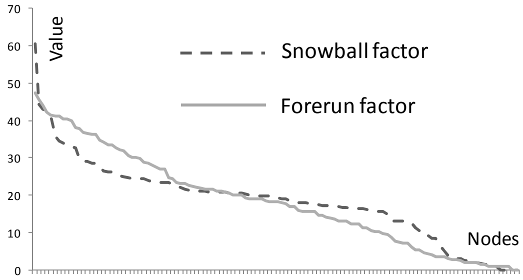
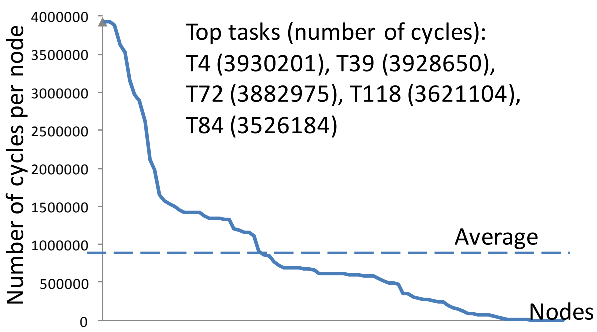


Figure 9. Snowball and forerun factor per task **Figure 10. According distribution for fig. 9**

The local cluster-coefficient shows how a tasks is likely to drive the clustering of tasks in the process. It is calculated as the quotient of existing edges to adjacent neighbors and the number of possible edges. Apart from five outliers with amaximum coefficient of one (T6, T33, T86, T88 and T111), the distribution shows a relatively linear behavior. Accordingly, those five tasks are connected to each possible neighbor, and close workgroups are likely to be necessary at this part of the process.



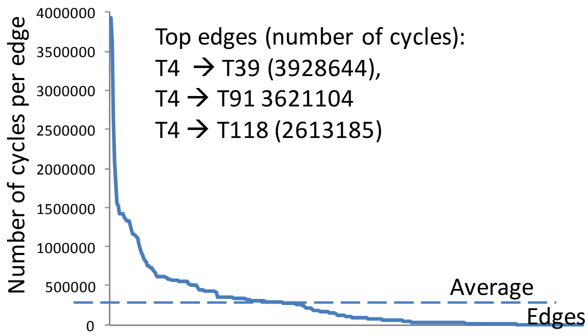


Figure 11. Number of cycles per node **Figure 12. Number of cycles per edge**

To assess iterations and uncertainty in the process, the metric number of cycles per node can give evidence. The more cycles take path via a node, the more this task will receive and distribute information from and for the overall network, and will therefore be of high influence. Figure 11 shows the distribution and the top values. Figure 12 shows how edges are involved in these cycles, pointing to important channels of communication. Both metrics describe the task T4 as most influential; while this coincides with e.g. the relevance as detected e.g. through the degree distribution, there is no direct correlation that can be obtained therefrom.

4.1.3 Metrics per module

On a different level, the use of structural metrics delivers meaningful results concerning the properties of modules involved into development processes as well as interdependencies among the different modules. Modules are predefined groups of entities, in contrary to clusters that develop during the process' progress caused by their intense interaction. In case of the analyzed process, 19 organizational units are predefined as modules.

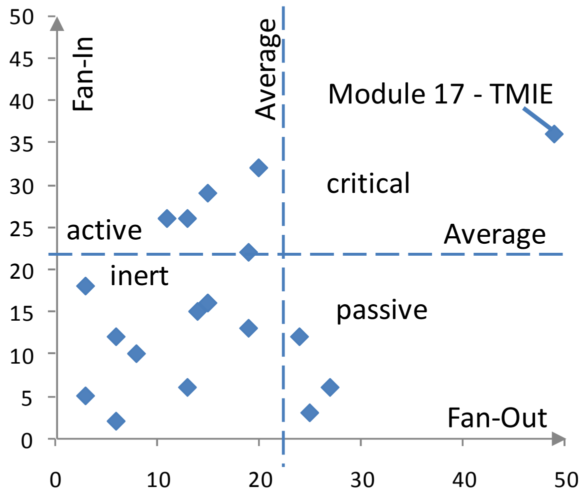


Figure 13. Fan-out and fan-in per module

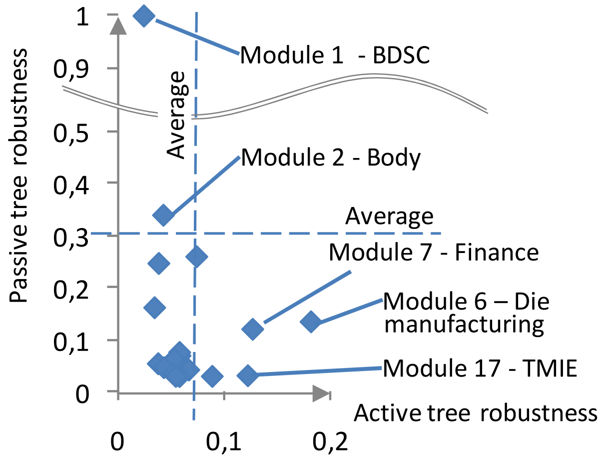


Figure 14. Tree-robustness

The fan-criticality (i.e. the number of outgoing and incident cross-border relations per module) allows comparing the out- and in-degrees of modules. The plot in Figure 13 shows module 17 stands out the most concerning both in- and out-degree, accordingly being most influenced as well as being the most influential module, being concerned with integration of a large set of components.

The metric tree-robustness is applicable either on complete domains or on modules. The portfolio of the values for active and passive tree-robustness (i.e. the quotient of the number of nodes with a nonzero value for snowball or respectively forerun factor and the sum of the according factors) in figure 14 shows modules 1, 2, 6, 7 and 17 as the most important outliers. This indicates that no module is dominated by both incoming and outgoing hierarchies of information flow, but that they either collect incoming information (all integration modules (6, 7, and 17) or generate it.

The distribution of the values of the global cluster-coefficient per module, i.e. the quotient of the sum of all local cluster coefficients and the number of nodes with an activity higher than one in a designated module, shows two outliers (modules 1 and 12). These modules are, therefore, the most likely to cause information transfers among the other modules.

The metrics delivering most information concerning the relation of internal and cross boarder flow of information of modules are module quality 1 and module quality 2. The metric module-quality 1 computes as the product of the number of edges that cross the border of the module and the number of edges within the module; module quality is calculated as the respective quotient. The first metric describes the flow of information through modules, while the second one describes the compactness of a module. For both modules, module 18 (TVIE – total vehicle engineering) can be identified as the most remarkable outlier.

Through these metrics the modules 1, 17, and 18 are determined as the most influential ones. In spite of the three more influencing modules, the process' organizational units are interconnected quite evenly. A rather low fraction of 122 (36,72%) of the 417 edges connect nodes within equal modules, characterizing the flow of information through the process as rather integrated among different modules. There is even one module without any internal connections (Module 15), for this reason it is positioned at the last positions for module quality 1 and 2 with value of zero. The importance of module 18 is rather logical because it contains most nodes per module. The importance of module 1 is a consequence of both of the process' start nodes being contained in this module, having high values for influence-describing metrics.

In summary, from a modules' point of view, the network can be described as well-balanced, with the modules 1, 17 and 18 having a higher importance because of their position in the process' progress at the beginning of the first phase or, respectively, at the end of the last phase.

4.2 Conclusions for the regarded process

For a better comparison, table 4 lists the core results. There, the influence measuring metrics are sectioned into active and passive ones. Active ones determine the distribution of information, the passive ones describe information sinks. For all metrics, the top-ten and the bottom-ten outliers are listed in the table. Start and end nodes are printed in bold there.

Concerning the structure of the analyzed development process, one result approved by every metric is the difference concerning the importance of the networks' two start nodes. Start node T1 is much more influential onto the overall process than T2 is, which is reasonable given that the design sketch generated in T2 only impacts tasks that are related to the exterior design of the car. Therefore, T1 ranks first three times, whereas T2 is not even once among the top ten of the active-influence measuring metrics, describing the inequality concerning their importance. This points, however, to the fact that the development process seems to be little design driven.

In general, among the top ten positions oft the active influence-measuring metrics many different tasks occur, showing a quite evenly distribution of importance. Not even do the process' start nodes occur among all top ten rankings, underlining the evenly distribution among the involved tasks. This result is consistent with the flow of information between the process' organizational units, which is likewise determined as homogeneous.

Some overall properties of the process can be deduced. Several indications categorize the development process as organized well-balanced and evenly: The two start nodes do have, logically, a high importance. But throughout the process, importance and influence onto the overall network is distributed among several tasks, a fact underlined by the high number of 65 (54% of all tasks) different tasks appearing among the top and lowest ten positions depicted in table 4.

Table 3. Top ten and lowest ten outliers for selected structural metrics

	Metric	Tasks arranged by position in ranking per value of metric																				
		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	…	111.	112.	113.	114.	115.	116.	117.	118.	119.	120.
Active	Activity	**T1**	T72	T11	T37	T3	T10	T17	T27	T61	T73	…	T108	T109	T110	T113	T114	**T85**	**T111**	**T117**	**T119**	**T120**
	Active reachability	**T1**	T3	T10	T11	T18	T25	T30	T37	T39	T29	…	T113	T5	T21	T23	T24	**T85**	**T111**	**T117**	**T119**	**T120**
	Active proximity	T114	T90	T89	T116	T83	T115	T12	T15	T87	T6	…	T113	T5	T21	T23	T24	**T85**	**T111**	**T117**	**T119**	**T120**
	Relative centrality	T39	T72	T37	T73	T4	T65	T91	T84	T112	**T118**	…	T6	T33	**T85**	T86	T88	**T111**	**T117**	**T119**	**T120**	**T1**
	Snowball factor	**T1**	T11	T10	T37	T3	T39	T16	T25	T27	T18	…	T113	T5	T21	T23	T24	**T85**	**T111**	**T117**	**T119**	**T120**
Passive	Passivity	T84	T65	T73	T72	T91	T92	T112	T96	T93	T94	…	T79	**T85**	T86	T88	T105	T113	T114	**T120**	**T2**	**T1**
	Passive reachability	**T119**	T118	T112	**T117**	T91	**T111**	T113	T114	T116	T104	…	T22	T36	T3	T5	T6	T8	T10	T21	**T1**	**T2**
	Passive proximity	T83	T114	**T111**	T82	T75	T104	T113	T115	T99	T98	…	T16	T22	T3	T5	T6	T8	T10	T21	**T1**	**T2**
	Forerun factor	T112	T91	T118	T73	T84	T96	**T119**	T94	T95	T93	…	T9	T36	T3	T5	T6	T8	T10	T21	**T1**	**T2**
	No of cycles per node	T 4	T 39	T 72	T 118	T 84	T 65	T 112	T 73	T 37	T 106	…	T 82	T 35	T 20	T 33	T 29	T 13	T 14	T 23	T 24	T 31

Another structural indication is the fact that within the entire process two tasks with only one incident and one outgoing node succeed each other only twice. All remaining connections (415 of 417) are edges connecting nodes with higher in- and out-degrees and are therefore far less critical. The number of edges being the only connection between two nodes (12 edges), is similarly low. The equally distributed interdependencies between the process' organizational units confirm the general properties from another point of view.

The small percentage (5.575%) of feedbacks among all connections also describes the flow of information as straight and evenly. However, there are about 4 million cycles that are, in particular, driven by T4; this conincides not only with its importance based on the degree but furthermore with its centrality (second outlier), confirming that especially here, decisions are taken and the core opinions are built about the product.

The results are partially consistent with the results of an earlier analysis [Braha & Bar-Yam 2007] speaking of few nodes being of high importance for the overall process. For single metrics, assessing single views onto the structure, this may be right. For example, task T39 ranks at first position of values for the metric relative centrality, appearing to be of outstanding importance. But, regarding all active influence measuring metrics simultaneously, it ranks at position 15, representing a rather high but not significantly outstanding importance.

Concerning another result of the earlier analysis, the newly calculated results are identical. Most nodes do have quite low degrees most connections within the network link entities with small activity and passivity. The metrics degree correlation based on edges and on nodes as well as degree distribution in section 4.1.2 confirms this result unequivocally.

5. Conclusion

5.1 Implications for industry

It could be shown that comprehensive process analysis is possible even without far reaching knowledge of a process to be determined. Already, from the basic information flow and the dependency network of tasks, core activities and their embedding into the process could be deduced, and it could be shown that different tasks have different characteristics in driving the process. Once this dependency network is available, the computation of the results takes, in fact, only a very short time; therefore, such an analysis can be run quickly for almost any process model.

The metrics can, therefore, be used to guide the set-up of a process improvement project in a more targeted manner; as it is possible, to extract process knowledge from large process charts, possible weak spots can be determined with little prior knowledge to generate hypotheses that can then be regarded in detail without wasting effort on tasks that, from a metrics' point of view, are little embedded and therefore have little impact onto the process.

The different patterns correspond, at the same time, to different interests of process management that have not been regarded here; however, the spectrum of results shows how some tasks are more focused on coordination (high degree), while others rather relate to systems architecting (centrality); a detailed discussion, however, is omitted here and shown in [Kreimeyer & Lindemann 2010].

These facets show how models can be used beyond current applications; this has been validated for the metrics using different other case studies [Kreimeyer 2010]. At the same time, the understandability of often complex process models is made more transparent by making explicit those patterns among its entities that guide the process. This helps not only improvement projects but also planning of future processes by identifying desirable patterns in existing processes that can, in a second step, be transferred to new processes.

5.2 Implications for research

A main conclusion concerning process analysis is that degree itself is not the only relevant foundation to reveal importance of entities in processes, although an important task is likely to have a rather high out- and in-degree. Rather it is necessary to regard the further progress of the incoming and outgoing paths of an entity. Much more, the active influence measuring metrics (i.e. activity, active reachability, active proximity, relative centrality, snowball factor) and passive influence measuring metrics (i.e. passivity, passive reachability, passive proximity and forerun factor) are able to give evidence, because not only the number of adjacent neighbors is counted, but furthermore the progress of outgoing edges throughout the process is regarded, describing the embedding of a task in the overall system "process". In the process, the tasks T72 and T73 rank among the top 10 values for both metrics, activity and passivity. But, considering the average position in the rankings of the active and passive influence measuring metrics, T72 ranks at position 26 or 31, respectively, and T73 at position 58 or 22, verifying a rather averaged importance onto the network as well as a rather averaged degree of being influenced by the network.

Concerning the detection of outliers, it was shown that an outlier must be numerically distant from the overall distribution (i.e. all metrics are comparative ones), but it must also be semantically relevant. This result confirms that a profound process analysis requires the application of a set of different metrics that support each other, in order to draw reasonable conclusion after the process has been focused on from different points of view and towards different patterns of complexity, such as e.g. the size, adjacency, paths, attainability and more (compare the categories in table 2). In turn, a set of metrics regarding different structural patterns can provide a much more balanced picture.

Concerning the use of structural metrics, it is essential to be able to compare generated values. One single value itself can give no evidence in general, as the metrics determine not the absolute quality of an entity in relation to an absolute scale. One possibility to make comparisons is to compare values of the same metric for all entities the metric was applied to, in order to find outliers. Another possibility is to see the generated values in comparison to the maximum value possible for this metric. This affects in particular metrics delivering one value for an entire domain. For example from the number of feedbacks a conclusion can only be drawn in comparison to the number of all connections within

the network. Another possibility is to compare properties of a process with properties of a random process having the same number of nodes and edges, as it was done in the earlier analysis of the process [Braha & Bar-Yam 2007].
The representation of metrics allowing an active and passive calculation, like active and passive reachability, can be presented most reasonable in a plot showing both values per entity. A characterization into the four fields inert, active, passive and critical, defined by the average value per axis, shows the difference in importance among the entities immediately. For a general analysis, active and the according passive values have to be regarded separately, in order to generate evidence concerning general active influence and general passive influence.

5.3 Future work

As shown, a set of metrics is able to provide a comprehensive picture about the qualities of a process available as a task dependency model; to this end, it is able to provide answers tovarious questions that are of interest in process management. A GQM based approach is currently being researched to provide an overall measurement system – to provide it, however, the interdependencies of the different metrics and how they support and relativize each other is necessary. A first indication can be drawn from the underlying characteristics that graph theory provides and that reflect in the categories shown in table 2. The results shown in this case study and the different foci that can be found (different tasks showing up as important for different metrics) ascertain this already. Yet, the precise relations among the metrics still need to be determined.
In a second step, the metrics can be applied to different networks – the common approach in process management is to relate tasks to each other via information flows or, more generally, their precedence relationships. Yet, other networks can be regarded, e.g. the documents or the dependencies of milestones across the tasks executed to fulfill the milestones. This has already been shown partially [Kreimeyer et al. 2008], yet the full use still needs to be demonstrated and consolidated.
Finally, the results the structural metrics can provide are rather high-level. Therefore, the coupling of them with other metrics can provide a more focused use of e.g. simulation approaches. Therefore, the results of the metrics need to be put into relation with results from such approaches. Different comparative studies are ongoing at present and will be published in time.

References

Bashir, A. H., Metrics for Design Projects: A Review, Design Studies Vol. 20, No. 3, (1999), pp. 263-277.

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

Cardoso, J., Approaches to Compute Workflow Complexity, Proceedings of The Role of Business Processes in Service Oriented Architectures, Leymann, F. et al. (eds.), IBFI Schloss Dagstuhl, 2006.

Kreimeyer, M., A Structural Measurement System for Engineering Design Processes, Dr.-Hut Munich, 2010.

Kreimeyer, M., Lindemann, U., A GQM Frameworkto Guide Process Improvement using Structural Analysis, Proceedings of 11th International Design Conference 2010.

Lindemann, U., Maurer, M., and Braun, T., Structural Complexity Management, Springer Berlin, 2009.

Kreimeyer, M., König, C., Braun, T., Structural Metrics to Assess Processes, Proceedings of 10th International DSM Conference, Kreimeyer, M. et al. (eds.), Hanser, Munich, 2008, pp. 245-258.

Mendling, J., Metrics for Process Models, Springer Berlin, 2008.

Newman, M.E., The Structure and Function of Complex Networks, SIAM, Vol. 45, No. 2, 2002, pp. 167-256.

Schlick, C. M.; Duckwitz, S.; Gärtner, T.; Schmidt, T.: A Complexity Measure for Concurrent Engineering Projects Based on the DSM, Proceedings of 10th International DSM Conference, Kreimeyer, M. et al. (eds.), Hanser, Munich, 2008, pp.219-230.

Dr.-Ing. Matthias Kreimeyer
Technische Universität München, Institute of Product Development
Boltzmannstr. 15, 85748 Garching, Germany
Telephone: +49.89.28915136
Telefax: +49.89.28915144
Email: matthias.kreimeyer@pe.mw.tum.de