

PROFILING PD PROCESSES BY COMBINING STRUCTURAL ANALYSIS AND SIMULATION

M. Kreimeyer, D. C. Wynn, P. J. Clarkson and U. Lindemann

Keywords: process, structure, metric, simulation, case study

1. Introduction

It is common in industry to redesign procedures and organisations periodically, sometimes as part of continuous improvement activities, sometimes as part of business process reengineering projects. Methodical approaches to systematically assess processes can help these efforts by helping to identify weak points and helping to find opportunities for improvement.

Research has generated different methods, models and analysis strategies to address this need. This paper shows how two approaches – structural metrics and simulation analysis – can be combined to generate insights additional to those which can be gained from either perspective in isolation. On the one hand, structural models and metrics can be used to provide a high level analysis of an existing network of tasks and intermediate documents (i.e. viewing the process as a task dependency model). On the other hand, stochastic simulation can be used to obtain an overview of how tasks interact over time by supplementing this information with estimated runtimes for each task. In this paper, both approaches are applied to a process model that originates from the body-in-white design process of a premium class sedan as used in a major German automotive manufacturer. We demonstrate and contrast both approaches to explore their strengths and weaknesses, used individually and in combination. We also weigh the effort required to build and analyse a model against the type and depth of results obtained through each approach.

The paper proceeds as follows. Section 2 provides background by reviewing approaches to systematic process exploration using structural and simulation analysis techniques. Section 3 introduces the case study on which this paper is based. Section 4 introduces the specific structural and simulation metrics that were considered and explains how they can be combined to generate additional insights. Section 5 applies both approaches, individually and in combination, to the process model developed during the case study. Section 6 discusses and reflects upon the ability of structural and simulation methods, individually and in combination, to provide insight into complex engineering design processes. Section 7 concludes.

2. Approaches to systematic process exploration

Process models, once constructed, can be analysed in different ways to draw insights about the processes they represent. This section provides an overview of different analysis approaches proposed in the literature. We consider approaches to analyse the structure of a process, in terms of the purposeful combination of tasks and their dependencies, alongside approaches which use simulation to analyse the time-varying behaviour of a process and the contribution of tasks within it. Furthermore we focus on use of these methods to identify possible areas for concern and opportunities for improvement; the results of such analysis are intended to focus the attention of domain experts who could then undertake further, more detailed investigation to identify specific improvements.

2.1 Structure-based methods for process analysis

The structure of a process is defined here as the patterns which emerge from the set of a process' entities (e.g., tasks) and the relationships between them. Structure-based methods analyse this interplay of entities to draw inferences about the behaviour of a process. Many methods of this type are based on the Design Structure Matrix (DSM); for instance, where algorithms such as banding, sequencing and triangularisation can be used to find a sequence of tasks expected to minimise iteration [Browning 2001].

Metrics to assess the structure of a system are most common in software engineering, mostly focused on quality aspects of software code with the goals of understanding, evaluating, controlling and forecasting the quality of a software program. Most of these metrics focus on evaluating the dependencies between parts of a software code, and are closely related to ideas of structural complexity. [Cardoso 2006] states that the criteria set out for software metrics also hold true for workflow metrics, and therefore that software metrics can be employed to evaluate certain aspects of processes.

[Cardoso 2006] compares a number of complexity metrics – namely Entropy from Shannon's Information theory, Kolmogorov complexity, McCabe's Cyclomatic Number, Cognitive complexity, and computational complexity theory – also concluding that they are applicable to measuring workflow complexity. He links the behaviour of a system to its structure by analysing log files that serve as protocols for both behaviour and the use of different control flows (i.e. structural patterns). Further metrics for workflows are introduced, among others, by [Gruhn and Laue 2006] (Lines of Code, McCabe's Control Flow Complexity, Nesting Depth, Jumps out of a Control Structure, and Cognitive Complexity Metrics). Such quality aspects are similarly regarded e.g. by [Mendling 2008], who identify possible errors and assess the degree of correctness of a model using a comprehensive set of structural metrics.

In engineering design, Summers and Shah (2003) develop a general measure of structural complexity, regarding the similarities and differences between design problem complexity, design process complexity, and design artefact complexity. Summers and Shah identify size, degree of coupling, and solvability as the three fundamental aspects of complexity, and measures for each aspect are defined. However, these measures are developed specifically for parametric and geometric problems. A set of structural metrics that focuses on engineering processes is proposed by [Kreimeyer 2010]. Kreimeyer collects structural patterns from the above authors into 52 different metrics. He shows how these can be used in conjunction with a generally-applicable process meta-model to comprehensively describe and assess the different aspects of a process.

2.2 Simulation-based methods for process analysis

Simulation has been used to explore PD processes and their behaviour for many years. Originating from the PERT models for the Polaris project in 1958 and later extended to the GERT method by NASA in the mid-1960s, these approaches have been extended and applied to many projects in industry, and to explore a wide range of problems in the academic literature.

Many different simulation approaches based upon models of activities and their relationships have been proposed in the engineering design literature. The simulation-based analysis used in this paper is based on the algorithm introduced by [Browning and Eppinger 2002]. This algorithm was chosen as, unlike many others, it does not require models to be 'well-structured', e.g., containing no cycles or deadlocks. It can thus be applied to any model which could be analysed using structural metrics and is suitable for application to models which were not originally constructed with simulation in mind. This was the case for the model constructed during the case study described in Section 3 and analysed in forthcoming sections.

3. A Product Development Process in the Automotive Industry

This section introduces the case study on which our analysis is based. The case study was part of a greater project to reengineer the collaboration process between embodiment designers and simulation engineers. For an overview, see [Deubzer, et al. 2007]. During the initial phase of the project, a process model was set up as the basis for assessing the structure of collaboration and other aspects

which are relevant to collaboration. This model is typical of those used for process management in industry. The model itself is described in greater detail in [Kreimeyer et al. 2007].

3.1 Overview of the process

The process modelling focused on the interaction between all embodiment design engineers and simulation engineers, both within the company and from service providers, who are involved in developing a car's body-in-white for serial production. The body-in-white comprises the body plus doors, hoods, and lids without further components (chassis, motor) and trim (windshields, seats, upholstery, electronics, etc.). In this context, simulation refers only to vibration, deformation, and air flow load cases.

The process which was modelled starts with the official launch of the project. It ends with the launch of production preparation after testing and the pre-series runs are finished, i.e. when design and simulation are no longer heavily involved. Thus, the process starts with a "customer need" which is delivered by the marketing department. It finishes after all components have reached the release level "ready for purchasing".

Different organisational units are involved in the process. The pre-development departments deliver the first input; the design and simulation departments only support the concept phase and become active during the prototype and serial development phase. After a decision has been made as to which concept is detailed for serial development after an initial prototype (commonly, elements of all different concept designs that still exist up to that point are combined into a final design), design and simulation departments complete the overall design, which is progressively tested and refined. All of this is supported by various external services in simulation and development and by suppliers – who do not only deliver the final components, but also support their development as integrated partners. In the process model, these three types of service providers are represented as archetypical organisational units. Overall, the process involves about 800 engineers in total.

3.2 Detailed model of the process

Figure 1 shows the model of this process on which the forthcoming analysis is based. The model is an adapted version of an ARIS EPC model, set up as a network of alternating tasks and business objects (strictly speaking, EPC prescribes an alternation of functions and events, with business objects modelled in addition; however, for clarity and because events and business objects often are very similar, the adapted modelling scheme was preferred by management when the model was constructed). Supporting IT systems and the organisational units responsible for every task are also captured in the model. Major project milestones appear as columns in Figure 1, creating swimlanes which collect the related business objects. The model was set up at a medium level of detail, i.e. each EPC task represents a work package of three to ten weeks' duration. On average, each task has a duration of six weeks.

The process model was assembled from data that was collected from various smaller process models and interviews with 68 engineers in all relevant departments across a time of approximately four months. The process as elicited from each individual interview was modelled, and the resulting partial model was fed back to the interviewee for verification prior to integration into the overall context. The integrated model was then discussed in a series of workshops with management personnel involved in the process. The model thus consolidates considerable knowledge on how the company operates.

The model comprises 134 different business objects that are processed concurrently by 160 different tasks. It further involves 27 different IT systems, 7 milestones, and 13 organisational units – three of which are external service providers. The model is structured using horizontal swimlanes for each organisational unit along an implicit left-to-right time axis that is not to scale. 11 sub-processes exist, shown as coloured boxes in Figure 1. Some of these take part internally and externally in parallel.

4. Combining structural and simulation analysis to explore process models

In this section, we describe some specific structural and simulation metrics and show how they can be combined for additional insights than can be gained through either in isolation. In Section 5 we present results from applying these metrics to the case study model.

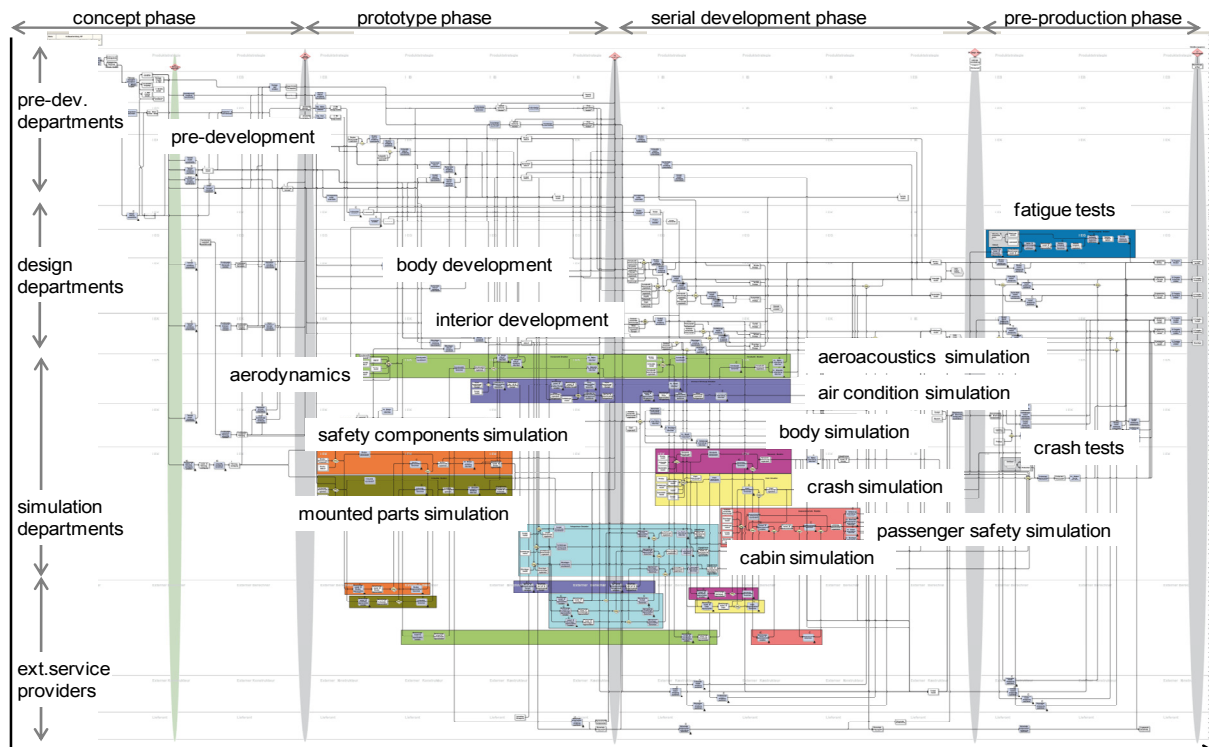


Figure 1. A model of the automotive design process

4.1 Metrics from structural analysis

As indicated in Section 2.1, many structural metrics can be applied to a process model; [Kreimeyer 2010] provides an overview presenting 52 different metrics in total. The metrics can be classified into six categories that represent different views of the process: 1) The role of individual entities; 2) sub-structures; 3) the global structure of the process; 4) its decision logics; 5) the quality of the process model; and 6) aspects of human cognition. These different metrics assess structural patterns regarding: the size and density of a process, the adjacency of entities, their attainability, their closeness, the hierarchies within it, the cycles among entities, the tendency for clustering, the connectivity of entities, the impact and interaction of several domains, and different paths across the process.

Only those metrics which assess individual entities in the context of the wider process are used for the analysis in this paper. The representative metrics used in this paper are intended to highlight important structural characteristics of engineering design processes by identifying nodes with unusual behaviour relative to others in the process, that could then be focused on as a possible concern or opportunity for improvement [Kreimeyer 2010]. Table 1 provides an overview.

4.2 Metrics from discrete-event simulation

Metrics from simulation concern the temporal aspects of a process, which cannot be studied using structural metrics alone. A key aspect of this is iteration in design. In a process such as that modelled in Figure 1, complex, nested and interacting iterations conceal the timing of tasks and information flows. As a result, it is not clear which tasks will be executed when, in which sequence, with what concurrency, and how often they may be revisited. The difficulty of fully understanding process behaviour from a structural model alone is further complicated by resource constraints, which can significantly influence when tasks are executed and revisited.

The main premise of this paper is that PD process simulation provides a way to supplement structural metrics with temporal information and thus improve the insights which can be drawn from the structure alone. In the following subsections, we first present the metrics drawn from simulation and show how they can be combined with the structural metrics to derive additional insights. We then

discuss the detail of the simulation approach and thereby describe how values for these metrics can be derived from real models such as that shown Figure 1.

Table 1. The structural metrics used in this paper

Metric	Description	Structural pattern
A(f): Snowball-factor B(f): Forerun-factor	<p>Snowball-factor: Total number of ‘downstream’ (preceding) nodes influenced by node f, weighted by inverse of shortest path length to root node f; Forerun: idem for upstream nodes</p> <p>Number of tasks in overall process that depend on this task; Large value = high propensity for change in task to propagate downstream Example: The root node has four nodes it reaches with path length one and four it reaches with path length two, therefore $E(\text{Root})=4*1+4*1/2=6$</p>	
C(f): Number of Cycles D(f) and E(f) Number of cycles and edges per node	<p>Number of cycles in which function f occurs: Propensity of task to be involved in iteration. Large value = many refinements required to get it right, no point making it ‘perfect’ first time. Also possible importance of communication to synchronise iterations and reduce rework.</p> <p>The counters D and E assess the range of distribution of cycle length for cycles in which function f or an edge between to functions participates</p>	

Calculation of metrics from simulation results

This section discusses how time-oriented metrics can be extracted from simulation results and presented a similar way to the structural metrics. The metrics can be derived from the results of any discrete-event simulation model which provides the execution profile of tasks in a process. We first describe the metrics in simulation model-independent terms, prior to describing the simulation we used to analyse the case study model.

All the metrics are based on calculations over the task execution profiles as revealed through simulation. For a given task and for one possible outcome of the simulated process, the profile expresses when that task is being worked upon (Figure 2). In the following notation, W_{nm} expresses the total number of attempts of task n in outcome m ; this varies from outcome to outcome and from task to task. For a given outcome m , task n ’s start times and finish times upon each iteration w can be expressed as $t_s(n, m, w)$ and $t_f(n, m, w)$ respectively. For the comparison in this paper, we consider the metrics shown in Table 2, which are each derived from $t_s(n, m, w)$ and $t_f(n, m, w)$ as revealed through simulation.

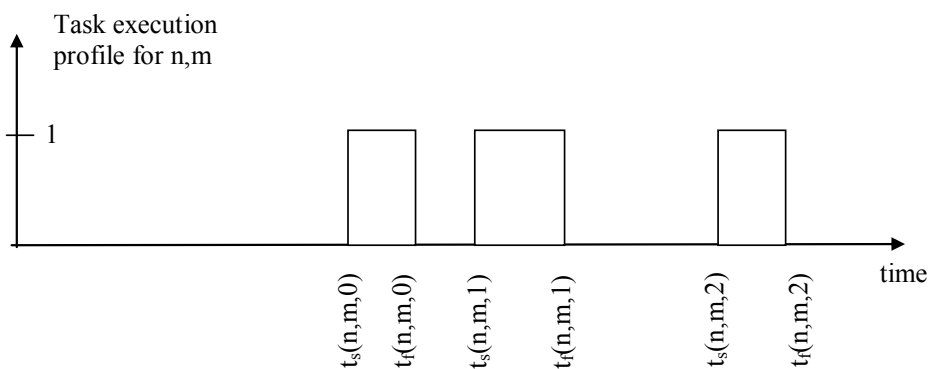


Figure 2. A task execution profile resulting from simulation

Table 2. Three simulation metrics

Description	Value for a set of outcomes m , size M
Expected start time of first attempt of task n	$t_s(n) = \frac{1}{M} \sum_{m=0}^M t_s(n, m, 0)$
Expected finish time of final attempt of task n	$t_f(n) = \frac{1}{M} \sum_{m=0}^M t_f(n, m, W_{nm})$
Expected number of iterations of task n	$i_\varepsilon(n) = \frac{1}{M} \sum_{m=0}^M W_{nm}$

4.3 Combining simulation and structural metrics

Structural and simulation metrics can be combined pair-wise in different ways. Some pairs of the simulation metrics and structural metrics are *complementary* – in that the combination may reveal additional insights over either metric in isolation. Other pairs are *comparable*, in that they might be expected to reveal a similar profile of objects in the process. Comparability could be useful as, for instance, it could allow approximations to simulation metrics to be computed through structural approaches, which require less data to be elicited. Some possible complementary and comparative relationships between the metrics outlined above are shown in Table 3. The remainder of this paper focuses on illustrating the benefits of analysing complementary metric pairs.

Table 3. Examples of comparison/combination of metrics

Simulation metric	Structural metric	Approach	Rationale
Expected start time of first attempt of task n	Snowball factor A for task n	Combination	SB factor is always bigger for tasks earlier in the process. Task start date allows comparison of SB factor for comparable tasks only.
Expected finish time of first attempt of task n	Forerun factor B for task n	Combination	As above
Expected number of iterations of task n	Number of cycles D in which function f occurs	Comparison	Number of cycles interpreted as ‘criticality’ of task

5. Applying the metrics to the car body development process

5.1 Calculating the structural metrics

The structural metrics were calculated by exporting the eEPC model of Figure 1 from ARIS and importing it as an MDM in the Looemo software, which provides functionality for calculating most of the metrics in Table 1. The remaining metrics shown in Table 1 were calculated using a spreadsheet.

5.2 Calculating the simulation metrics

The simulation metrics were calculated by analysing the same data set using the Cambridge Advanced Modeller software (formerly P3) [Wynn 2007]. Since several process simulation codes are provided by this software and could serve as the basis for calculating the metrics of Table 2, an algorithm was selected by considering the properties of the model and of the underlying process.

Selection of a simulation model

As indicated in Section 2.2, several different process simulation methods are available, and each is appropriate to different situations and models. To identify the most appropriate approach for this analysis, the model of Figure 1 was first visualised as a DSM as shown in Figure 2. The matrix shows

only activities in the process; it is sequenced to minimise the number of dependencies above the leading diagonal, which each represent possibilities for iteration to occur.

The DSM clearly indicates a complex knot of feedback and feed-forward of information occurring around the centre of the process. Since no information was available regarding the procedure for resolving these iterations, this ruled out use of a precedence-oriented simulation technique, such as PERT-based approaches. Furthermore, since information flows were explicitly defined, and no knowledge was available regarding the contribution of tasks to levels of knowledge of the product, a rule-based approach such as Signposting was not suitable either.

We therefore based the analysis on the combined precedence-dependency simulation algorithm proposed by [Browning and Eppinger 2002]. This algorithm has been applied to industry case studies, and is accepted by other authors as the basis of their simulation approaches. Hence, it provides a good basis for investigating the car body development process and for demonstrating our combined metrics approach.

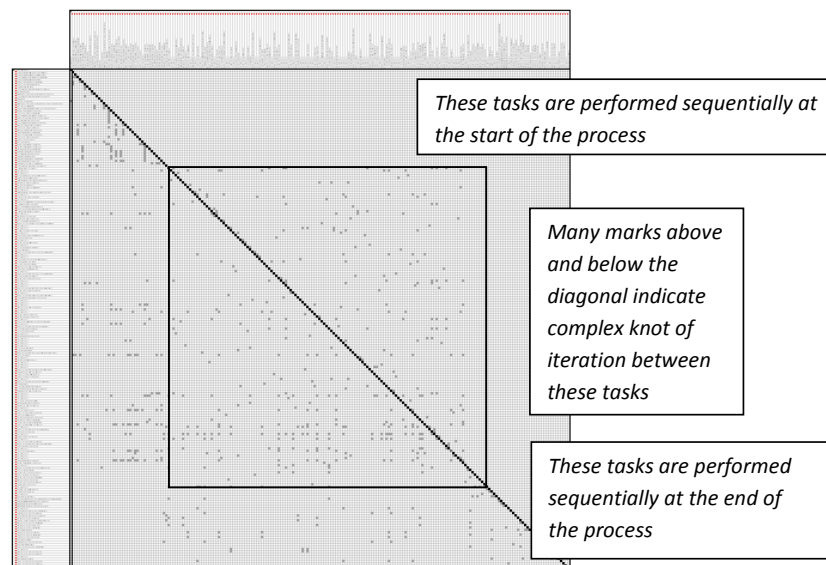


Figure 3. The process model visualised as a Task DSM

In Browning and Eppinger's model, tasks in a DSM are specified in their sequence of execution which is referred to as the architecture of the process. Therefore, a dependency in the upper diagonal of the matrix indicates that a task attempted later in the sequence feeds information back into a task that was previously completed. Browning and Eppinger's model conceptualises this as a change which could occur to the input data of the upstream task when the downstream task is completed. It is assumed that if this occurs, the upstream task will require rework to absorb the change. In turn, the output information from the upstream task could change – potentially generating knock-on rework to other tasks. Rework accumulates on top of existing work to be done, such that the work remaining on a task can increase while that task is still in progress. This can be viewed as 'churn' iteration caused by information exchange between concurrent, overlapping tasks. If a task is not in progress during rework, the task and its successors must be re-attempted in a new 'distinct iteration'.

In Browning and Eppinger's model, each task has cost and duration specified as a triangular probability density function (TriPDF). Each dependency between two tasks is supplemented with likelihood and impact values. For a given dependency above the diagonal, likelihood indicates the probability that when the task in the column is completed it will add rework to the task in the row, which requires information from it. The impact specifies the amount of rework required, specified as a multiple of that task's original cost and duration. Browning and Eppinger's model assumes that the impact will always be less than or equal to 1, ie. that rework caused by change in a task's input information will never require more effort than the original work. Likelihood and impact values below the leading diagonal govern knock-on rework in the model – how likely that rework will propagate from the first revisited task to the next task downstream, and so on.

Parameterising the simulation

To simulate the car body development process, using this algorithm, we assumed that each task has a duration with triangular probability density function, having best-case value of 3 weeks, worst-case value of 10 weeks, and most likely value of 6 weeks. These parameters reflect the known granularity of the original model, as described in Section 3.2. The baseline sequence of tasks, which can have a significant effect on the amount of rework in the simulated process, was taken to follow the optimal sequence shown in Figure 3 which broadly reflects the ‘time’ axis of Figure 1. All likelihood and impact values were assumed to be 0.5, since no actual likelihood/impact information was available. The numerical results of the simulation could not be validated in detail, but appeared reasonable given the known duration of the real process.

5.3 Results

Task profile through structural metrics

Counting all possible cycles in the process chart showed that, in total, 39,105,321 different cycles exist across both tasks and business objects. Figure 4 (left) shows how cycles of different lengths are distributed in the model (each pairing of a task and business object counts as length 1), and indicates the business objects which appear in the most cycles. The results suggest that while there are very few short cycles, which might indicate iteration localised among a few entities, cycles of medium lengths, involving 15 to 25 tasks (and thus business objects) are significant in the process.

The table in Figure 4 (right) shows the business objects which appear most often in cycles across the model. For the case of the crash simulation results, these results could indicate that a core team which manages this data has a strong influence on how smoothly the process runs. Possibly, if that team were able to influence the crash results such that all partners agreed on an early exchange of possibly immature information, some unnecessary iterations might be prevented.

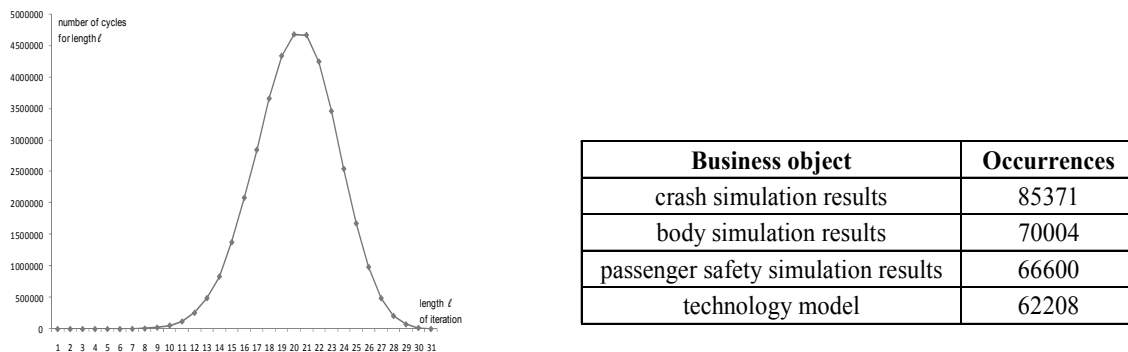


Figure 4. Distribution of cycle lengths (left) and business objects found in most cycles (right)

Table 4 shows the information transfers between tasks and business objects which participate in the most cycles. These information transfers are represented as edges (arrows) in the model of Figure 1.

Table 4. Occurrence of control influences of tasks on business objects within iterations

Control influence (task → business object)	Occurrences
coordinate crash simulation → body simulation results	69839
set-up model for crash simulation → simulation model for crash	53904
coordinate body simulation → body simulation results	50070
Coordinate passenger safety simulation → passenger safety simulation results	46916
design concept of cockpit → concept of cockpit	33952

This reveals that tasks which generate the important business objects revealed in Figure 4 also appear frequently in cycles. For instance, the strong influence of activities and business objects involved in

crash simulation is easily visible. However, the distribution between tasks which coordinate and tasks which generate information is uneven.

Unlike in Figure 4 (right), setup of the technology model is not among the top five elements of Table 4. Further analysis revealed that this is due to the fact that the technology model is drawn from many sources, with the result that its formal set-up is not as relevant as other tasks. At the same time, the concept design of the cockpit appears as an important driver in iterations, which is not shown in Figure 4 (right).

Whereas the technology model is among the top four objects shown in Figure 4 (right), Table 4 reveals that it is much less involved in cycles, and therefore iterations than the crash results, even though the technology model is designed as a central means of coordinating all design efforts by collecting all relevant core measures and information. This suggests another possibility for more detailed analysis to explain the discrepancy and potentially improve the process.

Task profile through simulation

The model was then simulated using the algorithm of Section 5.2. Figure 5 (left) shows the histogram of duration (increasing across the page) vs. cost (increasing out of the page) of the simulated process, evaluated using the Cambridge Advanced Modeller (CAM) software. Figure 5 (right) shows the profile of tasks in the process. Each point on the plot shows a different task. The axis up the page shows the mean number of *distinct* repetitions of each task. The axes out of the page and left-to-right across the page show mean first start time and mean last finish time of each task.

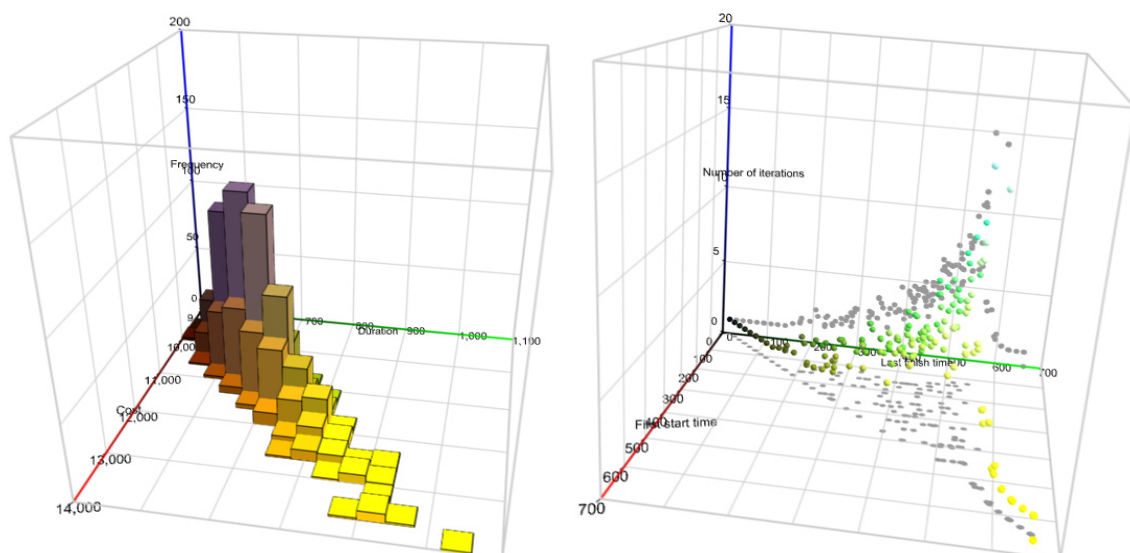


Figure 5. Process duration and cost (left), profile of tasks through simulation (right)

The right-hand plot of Figure 5 allows tasks with certain behaviours to be clearly identified. Considering the ‘floor’ of the plot, the leading diagonal represents the timeline of the process. Tasks off the leading diagonal tend to have long lead time - either because they are heavily involved in concurrency and ‘churn’ rework due to concurrent exchange of information (off-diagonal tasks close to the ‘floor’), or because they are completed and then revisited many times in distinct iterations, due to mistakes discovered after they are finished. The topology of the right-hand plot reveals that tasks tend to be involved in one or the other iteration modes, but not both. This suggests that tasks heavily involved in ‘churn’ iteration might be improved by facilitating information transfer to and from those tasks; whereas tasks involved in repetition could indicate potential for improvement by identifying the ultimate cause of that rework, which is initiated elsewhere in the process.

Combination of the metrics

To illustrate how additional insights can be derived from combining pairs of structural and simulation metrics, this section shows one way in which simulation can add value to structural metrics; and subsequently suggests how structural metrics can add value to simulation studies.

Supporting structural analysis through simulation: The *snowball factor* and *forerun factor* metrics were combined with *expected start time of first attempt of task* and *expected finish time of last attempt of task*. The results are shown in Figure 6. This comparison allows the structural metric results to be compared only for those tasks which are performed around the same time in a process. This leads to a more meaningful result, as it can be expected that the snowball factor for tasks earlier in the process will be significantly greater than those performed later, and as a result it is only meaningful to compare this metric across tasks executed around the same time. In other words, the tasks in a column with high snowball factor are more critical for the downstream process than those with lower snowball factor – and comparison between the snowball factors of tasks in different columns should be approached with care.

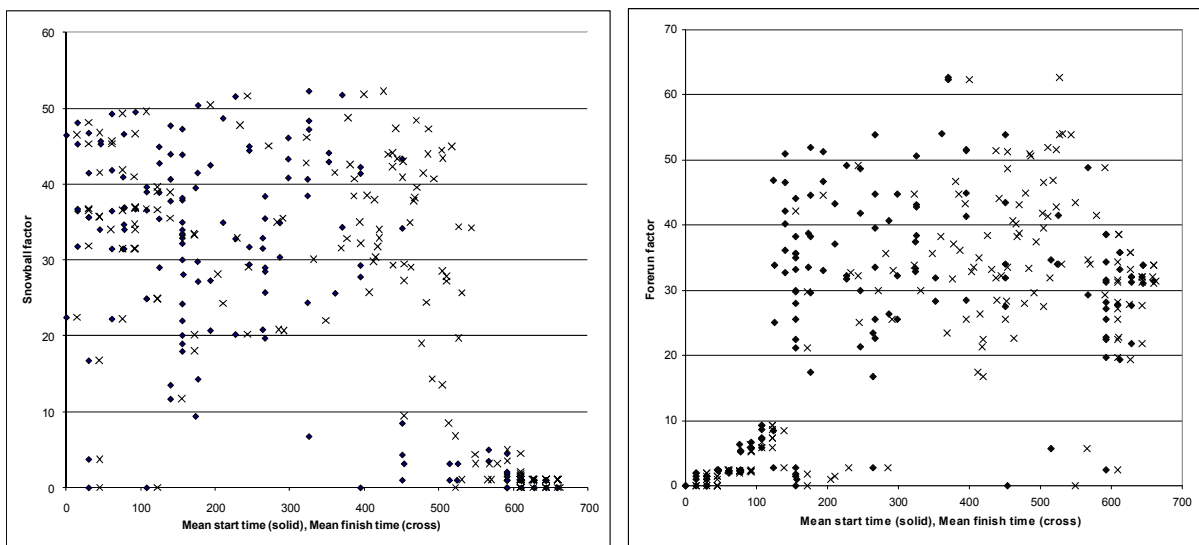


Figure 6. Illustrative combination of structural and simulation metrics

Supporting simulation through structural analysis: A second possibility for combination is to use structural metrics to assess how the impact of each task upon the process will change for different process architectures. As a process is reorganised, which is represented as resequencing the order of attempting tasks in Browning and Eppinger's [2002] simulation, the simulation results will change; yet the structural metrics will remain the same for each task since the structure of information flows is not influenced by the sequence of attempting tasks. Thus, process simulation can be used to identify a better architecture which results in lower expected cost and duration, while the combination of structural and simulation metrics as shown in Figure 6 allows the importance of each task to the reorganised process to be identified (the vertical position of each point, representing a task, will remain constant with the activity sequence although the horizontal position will change). This could be used to highlight how the 'most critical' tasks are likely to change when the process is reorganised. In turn, the combined analysis could highlight potential problems with the proposed new process, and indicate possibilities for further improvement by improving newly-critical tasks which would not be revealed through simulation alone.

Further analysis revealed that the identification of tasks with *extreme metric values* (such as the tasks occurring the most times in cycles vs. the tasks involved in the most distinct iterations) is roughly comparable, yet not identical in ranking between the structural and simulation approaches. Further work is required to identify the source of these differences and thus to assist in interpreting the insights gained through the different approaches.

6. Conclusion and further work

Metrics based on structural and simulation analysis of process models offer two ways to study process models to understand how the processes they represent could be improved. Various authors have discussed one or other of these types of analysis; this paper has contributed to the discussion by showing how these metrics can be combined to generate additional insights, based on analysis of a model of an industry design process of realistic complexity. The analysis highlights how the complex model developed through the case study can be understood in different ways through application and combination of different metrics.

Through this case study, the paper reveals some general strengths and weaknesses of the structural and simulation-based analysis approaches, which have previously been examined only in isolation. Structural metrics reveal no absolute properties of tasks or business objects, but only their relative properties compared to others in the same process. On the other hand, simulation provides a more detailed estimate of values related to the scheduling of tasks, which is not possible through structural metrics. A more balanced picture can be gained through combination of the approaches, and a combined approach can help make more informed decisions about process improvement measures. A rich picture cannot be developed from just one or very few metrics alone, regardless of whether those metrics are simulation or structure based.

Another notable difference between the approaches is the computational expense of their evaluation. Simulation-derived metrics are easier to compute than graph-based metrics; those used in this paper required less than a minute to compute on a desktop PC including the simulation itself, whereas the structural metrics took several hours. On the other hand, simulation-based metrics do require more input information and assumptions than structural metrics, which require only flows of information between tasks. This data is often already available in process maps within a company.

In conclusion, the two process analysis approaches studied in this paper appear to be complementary, revealing different characteristics of process execution. In particular, propagation of errors and distribution of information is revealed using structural metrics, whereas timing and scheduling can be explored through simulation metrics. The more detailed process data required for simulation appears to generate a deeper understanding; yet based on this study it is not possible to conclude whether this deeper understanding is a more accurate reflection of the process in reality. Finding a way to compare the quality of findings from each approach is a significant area for further research.

In terms of practical insights, our analysis revealed that structural analysis is most useful for high-level exploration which can be used to decide on a detailed analysis strategy, which could possibly involve additional knowledge elicitation. Structural metrics can also be used to obtain foci for possible risks in process planning to document existing procedures to facilitate continuous improvement and focused learning. When numerical data is available, simulation may yield more directly-applicable results (e.g., through generation of Gantt charts of optimised processes).

More generally, the potential benefits from combining two distinct and formerly unrelated research strands was confirmed – in general the results are complimentary and can be used in conjunction to obtain more complete picture. Further benefits could be gained through the transfer and comparison of models within the research community – the comparatively low effort in this case showed that the bridging between different research projects and software tools is relatively easy and has potential to yield interesting insights. In future we aim to explore questions of metric comparability through analysis of multiple datasets on available real process models from different companies in the automotive industry, which we are currently preparing.

More work is required to understand whether the different types of metric give comparable results on different models, and if not why so. Of particular interest is the comparison of *cycle* based structural metrics to simulation metrics. Cycle-based structural measurement is similar to simulation-based metrics which concern the number of iterations a task is involved in, therefore a reasonable correlation is to be expected. It would be especially interesting to know whether simulation will yield the same results as structural metrics if all tasks were similar in timing – this would answer the question to what level of detail the behaviour can be inferred from the structure. However, at time of this research no sufficient algorithm for the computation of cycles on a model of significant complexity was available.

References

- Browning, T. R., "Applying the design structure matrix to system decomposition and integration problems: A review and new directions", *IEEE Transactions on Engineering Management*, 48, No.3, year, pp. 292-306.
- Browning, T.R. and Eppinger, S.D. "Modeling Impacts of Process Architecture on Cost and Schedule Risk", *Engineering Management, IEEE Transactions on*, Volume 49, No. 4, 2002.
- Cardoso, J., "Approaches to compute workflow complexity", *Proceedings of the The Role of Business Processes in Service Oriented Architectures*, F. Leymann, et al. (Ed.), Schloss Dagstuhl, Germany, 2006.
- Deubzer, F., Kreimeyer, M., Herfeld, U., Lindemann, U., "A strategy for efficient collaboration in virtual product development environments", *Proceedings of the International Conference on Engineering Design, ICED'07, Paris, 2007*.
- Gruhn, V., Laue, R., "Complexity metrics for business process models", *Proceedings of the 9th International Conference on Business Information Systems, BIS 2006, Klagenfurt, Austria, 2006*.
- Kreimeyer, M., "A structural measurement system for engineering design processes", *Dr.-Hut Munich, 2010*.
- Kreimeyer, M., Eichinger, M., Lindemann, U., "Assessment of process networks using graph and network theory based key figures", *Proceedings of the FUBUTEK, D. Van Velden (Ed.), Delft, The Netherlands, 2007, pp.13-19*.
- Mendling, J., "Metrics for process models", Springer Berlin, 2008.
- Summers, J., Shah, J., "Developing measures of complexity for engineering design", *Proceedings of the DETC'03 ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, S. K. Gupta (Ed.), Chicago, 2003, pp.12.
- Wynn, D. C., 2007, "Model-based approaches to support process improvement in complex product development", *Ph.D. thesis, University of Cambridge, Cambridge*.

Dr. Matthias Kreimeyer
Institute of Product Development, Technische Universität München
Boltzmannstr. 15, 85748 Garching, Germany
Telephone: +49.89.28915136
Telefax: +49.89.28915144
Email: matthias.kreimeyer@pe.mw.tum.de
URL: <http://www.pe.mw.tum.de>