

DIFFERENT LEVELS OF PRODUCT MODEL GRANULARITY IN DESIGN PROCESS SIMULATION

Maier, Jakob F. (1); Eckert, Claudia M. (2); Clarkson, P. John (1)

1: Cambridge University, United Kingdom; 2: The Open University, United Kingdom

Abstract

The design of many products is incremental, based on a prior architecture and may be thought of as a series of changes to an existing design. Nonetheless, design changes and their propagation complicate design process planning. They are a major source of rework and lead to frequent rescheduling and re-prioritisation of design tasks. Few methods exist to estimate the impact of such planning decisions with regard to both the design process and the product. Process simulation may provide support in this context when it is based on an appropriate description of the product. However, it can be hard to determine a suitable level of description, or granularity, for the product model. This article explores, by reference to a diesel engine, the implications of product model granularity for simulating design change propagation, and design process planning.

Keywords: Design process, Simulation, Model granularity, Design Structure Matrix, Engineering change

Contact:

Jakob F. Maier
Cambridge University
Engineering Design Centre
United Kingdom
jfm45@cam.ac.uk

Please cite this paper as:

Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

Planning design processes is a difficult endeavour throughout a product development project. It encompasses activities ranging from strategic planning decisions related to system architecture selection to operational scheduling of design task sequences. Often, designers need a quick way to estimate design effort based on the current state of the product, for example when selecting among system architecture alternatives, planning an iteration loop or choosing a way to implement a change to respond to an unexpected problem or a new customer request. The design tasks necessary to carry this out differ in their nature. While some tasks like testing are similar regardless of the solution chosen, others depend on the selected alternative or on what exactly needs to be changed. Analysing the effects of changes on both product and process enables the designer to consider process effort as soon as a modification to an existing product is considered and before making detailed decisions about the product or the process. This paper brings together three different modelling approaches: (1) Change Prediction Method (CPM) (Clarkson et al., 2004) which assesses the risk of change propagation; (2) aggregation of change prediction models (Ariyo et al., 2007); and (3) process simulation based on change prediction models (Maier et al., 2014). It compares process simulations of a detailed CPM matrix with an aggregated matrix to assess the effect on the predicted process effort and shows that more abstract matrices can also be used to start the process planning by accounting for the level of granularity.

The following section briefly discusses model granularity in general and with relation to product decomposition. Section 3 introduces the simulation model used to conduct the experimental part of in the context of the broader simulation literature, while Section 4 introduces the algorithmic approach to calculate likelihoods of change propagation on different levels of granularity and the product model used in this study. Section 5 describes the experimental set-up, analyses and compares the simulation results on two levels of granularity and reflects on the implication of different levels of granularity for the simulation model, before conclusions are drawn in Section 6.

2 MODEL GRANULARITY AND DECOMPOSITION

The Oxford Dictionary (2014) defines granularity as “The scale or level of detail present in a set of data or other phenomenon”. Different levels of granularity usually imply hierarchical models, which relate different levels of detail to each other. In the process of hierarchically decomposing a product structure, larger systems are divided into smaller subsystems and the relationships between these subsystems are mapped. Simon (1962) distinguishes between interactions among subsystems and interactions within subsystems and notes that the latter are generally stronger. He also states that hierarchical systems are often ‘nearly decomposable’, which means that the interactions between its subsystems may be weak but not negligible. This highlights the importance of system decomposition and thus the resulting level of granularity of its subsystems has for the analysis of product architecture and connectivity. Different tasks and different stakeholders involved in the design process demand or require information on different levels of abstraction. Additionally, the choice of granularity can influence the time or cost to build, maintain and use a model. On the other hand, a certain level detail may be necessary to satisfy the client’s demands or to obtain results with sufficient accuracy or fidelity. As a result models often have uneven levels of detail, where some better-known issues are presented in a great amount of detail while others are left at an abstract level.

Although modellers frequently have to make choices regarding decomposition, Chiriac et al. (2011) note that past research does not discuss the effects of the level of granularity on architectural analysis. Chiriac et al. (2011) focus on how the degree of modularity of a product is affected by the level of granularity it is represented in. A range of idealised systems and one real-world complex system are represented in Design Structure Matrices (DSMs) on two different levels of granularity and analysed for modularity. They conclude that the results of architectural analysis can be distorted by the level of granularity of its components and advise to be cautious when defining system decomposition for analysis. The purpose of this paper is thus to contribute to this discussion and add a different perspective by investigating the implications of model granularity, and in particular model decomposition, on the simulation of change propagation and rework.

3 DESIGN PROCESS SIMULATION BASED ON PRODUCT MODELS

Process simulation models usually describe process tasks, their dependencies and probabilities of failure (for a review see Wynn, 2007). While a small number of integrated product and process models exist, they often do not use product specific information for process simulation.

Maier et al. (2014) present a discrete-event simulation model to study the effects of change propagation and resulting rework by generating an array of possible activity sequences based on a product model. The model is based on a DSM of components and their interfaces in a design, in which numeric entries between 0 and 1 describe the impact and likelihood of change propagation associated with each dependency (Clarkson et al., 2004). Design maturity is modelled on a component level and represents the state of design progress. The model synthesises the effects of three important issues in design: iteration carried out to progress the design (Smith and Eppinger, 1997), iteration necessary to correct errors or address design changes (referred to as rework in this article (see e.g. Wynn et al., 2007)), and change propagation due to structural interdependencies in the product being designed (Clarkson et al., 2004). Simulation experiments based on this model can help to evaluate simple task prioritisation rules based on factors like component connectivity, maturity and development times (Table 1). Maier et al. (2014) analyse the performances of a range of task prioritisation policies for various input product models. While they suggest that certain prioritisation rules are beneficial regardless of the scope of the project and of various other contextual variables, the question remains how the granularity of the input model affects the simulation and its results.

Progress is represented as transitions between maturity levels, which are assigned to each component to describe its state of progress throughout the simulation. Five discrete levels of maturity are used, allowing for four transformations between levels. Work on a component increases its maturity level. A decrease can be triggered by rework or by propagated change. In the case of change propagation, the corresponding entry in the impact DSM determines by how much the maturity level is reduced.

The approach makes a number of simplifying assumptions:

- The product architecture, as represented by the impact and likelihood DSMs, can be modelled in advance of the design process and does not change during it.
- The simulation starts with components having the lowest possible maturity and ends when all components reach their maximum maturity levels.
- Tasks that are similar for all solutions, such as testing or documentation are factored out. Evolution of maturity levels towards their final values thus reflects progression in designing the individual components.
- Interconnected components have to progress together. A component can only be selected for further work if the maturity levels of all other components it is dependent on are at most one level lower than its own. Thus, in the absence of rework the maturity of any component would never be more than two levels higher than any of the components it is dependent on.
- Reaching each maturity level for the first time requires an identical percentage of the total duration assumed to be necessary to complete the component.
- Task durations depend on change magnitude and are subject to learning effects (see e.g. Browning and Eppinger, 2002). The improvement curve used is based on a concept by Cho and Eppinger (2005), assuming an improvement by a certain percentage (25%) on each consecutive attempt until a minimum fraction (25%) of the original duration is reached.
- Task durations are not subject to probabilistic variation.

Change propagation is modelled based on the logic of the CPM (Clarkson et al., 2004) and simulated using a Monte-Carlo approach governed by the likelihood DSM. Propagation of changes is terminated after five steps, noting that this limit has been found realistic by Pasqual and de Weck (2011) and that the multiplying of probabilities make long propagation chains very unlikely. The propagation path ends if the algorithm revisits a component already selected for a change in the current stage.

In the model, changes are triggered randomly with a constant probability of occurrence. Changes can be interpreted as being caused directly by the task or as uncovering a problem originating elsewhere. If a change occurs it is taken into account by reducing the maturity level of the affected component. If a change occurs in a component that is being worked on, the activity is interrupted. This reflects that when the need for a change is discovered, it causes a step backwards in the design process and hence reduces the maturity of a component; rework is needed to attain its prior state.

Priority decisions occur every time a task is completed and select one of several possible tasks to continue. The decisions are simulated using priority policies consistently prioritising the task with either the minimum or maximum value for the selected decision criterion (see Table 1 for a numbered overview). If two or more tasks have identical values, a random tiebreaker chooses between them.

Table 1. Decision policies and corresponding criteria of components to be prioritised.

Decision criterion	Prioritisation rule	
None	0: Random selection	
Task duration	1: Shortest first	2: Longest first
Current maturity level	3: Lowest first	4: Highest first
Active sum in binary DSM	5: Lowest first	6: Highest first
Passive sum in binary DSM	7: Lowest first	8: Highest first
Active sum in risk-DSM	9: Lowest first	10: Highest first
Passive sum in risk-DSM	11: Lowest first	12: Highest first
Active sum in impact-DSM	13: Lowest first	14: Highest first
Passive sum in impact-DSM	15: Lowest first	16: Highest first
Active sum in likelihood-DSM	17: Lowest first	18: Highest first
Passive sum in likelihood-DSM	19: Lowest first	20: Highest first
Total attempts	21: Fewest first	22: Most first
Total amount of rework	23: Lowest first	24: Highest first

4 CHANGE PROPAGATION ON DIFFERENT LEVELS OF GRANULARITY

4.1 Change propagation on different levels of granularity

Extending the CPM, Ariyo et al. (2007) present an algorithmic approach to predict change propagation on coarser levels of granularity through a bottom-up aggregation procedure. They distinguish between intra- and inter-system connectivity. Based on that, their algorithm allows to calculate likelihoods of change propagation from system-to-components, components-to-systems and systems-to-systems. Being able to calculate change propagation likelihoods on different levels of granularity can reduce the effort to develop hierarchical models (once the initial fine DSM is created) and allows for consistent estimation of change likelihood on multiple levels. However, the approach also has some limitations. For instance, it is not possible to estimate the impact of change propagation on different levels. It can be expected that the impact of a change propagating from a system to another system is higher than from a component to a component but the approach does not enable such an assessment. A measure of change impact can help to estimate the magnitude of rework required after a change. However, there is no indication of how the process might be affected by these changes. Also, intra-system change propagation is not represented anymore on a system level, which, in the case of a highly modular product, could conceal that amount of change propagation and rework in a design. Section 5 describes how a better understanding of these effects can be gained by simulating the process of designing the respective product, including change propagation and resulting rework.

4.2 Used product model with two levels of decomposition

For the purpose of this study, we use the product model of a diesel engine described by Jarratt et al. (2004) and adopted by Ariyo et al. (2007). Details about the data capturing can be found in the original publications. While other product models and hierarchical decompositions can be used in the simulation model, this article focuses on one product and two levels of granularity to illustrate the underlying principles. Figure 1 shows the component level DSM of the Diesel Engine, which comprises 41 components. This matrix shows the direct likelihoods of changes propagating from one component to another with the cells shaded accordingly. The DSM also highlights which components belong to which system. Ariyo et al. (2007) note that the specific decomposition depends on the purpose as well as the context and requires some negotiation. For the purpose of their study, they group the 41 components into 10 systems. This decomposition is adapted here.

	Combustion System	Transmission	Control and Electricals	Fuel System	Lubrication System	Start. Syst.	Ignition System	Air Intake	Cooling System	Ex
Combustion System	Cylinder Head Assembly Cylinder Block Assembly Piston Rings Gudgeon Pin Conn Rod Lifting Eyes	Crankshaft Main Bearings Adapter Plate / Flywheel Housing Flywheel Ring Gear Gear train Gear Driven Auxiliary (compressor) Balancer Belt Driven Auxiliary (hydraulic pump)	ECM Crank Pulley Damper Belt Alternator Bracket Wiring Harness	High Pressure Fuel Pipes Fuel Pump Fuel Injection Assembly Low Pressure Fuel System Fuel Filter	Sump Oil Filler Engine Breather Oil Pump Oil Filter Oil Cooler	Starter Motor Starting Aid	Valve train Cam Shaft Push rods Timing Case	Turbocharger Aircharge Cooler Air Intake Air Filter	Fan Drive Fan Extension Coolant Pump	Exhaust Manifold
Combustion System	.5 .1 .4 .2 .1	.1 .4 .1 .1 .1 .2 .2 .1	.1 .1 .1 .3	.2 .2 .5 .2 .1	.1 .1 .1 .1 .1 .1	.2 .4 .2 .2 .2	.1 .1 .1 .1 .1	.1 .1 .1 .1 .1	.2 .1 .1 .1 .1	.2 .1 .1 .1 .1
Transmission	.3 .3 .2 .1 .1	.2 .3 .3 .1 .1 .2 .2 .3	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2
Control and Electricals	.2 .3 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2
Fuel System	.1 .3 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1
Lubrication System	.2 .1 .3 .2 .1 .1 .1 .1	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2
Start. Syst.	.3 .3 .3 .3 .3	.3 .3 .3 .3 .3 .3 .3 .3	.3 .3 .3 .3 .3 .3 .3 .3	.3 .3 .3 .3 .3 .3 .3 .3	.3 .3 .3 .3 .3 .3 .3 .3	.3 .3 .3 .3 .3 .3 .3 .3	.3 .3 .3 .3 .3 .3 .3 .3	.3 .3 .3 .3 .3 .3 .3 .3	.3 .3 .3 .3 .3 .3 .3 .3	.3 .3 .3 .3 .3 .3 .3 .3
Ignition System	.2 .1 .3 .2 .2 .1 .2 .1	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2
Air Intake	.2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2	.2 .2 .2 .2 .2 .2 .2 .2
Cooling System	.1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1	.1 .1 .1 .1 .1 .1 .1 .1
Ex	.4 .4 .4 .4 .4	.4 .4 .4 .4 .4 .4 .4 .4	.4 .4 .4 .4 .4 .4 .4 .4	.4 .4 .4 .4 .4 .4 .4 .4	.4 .4 .4 .4 .4 .4 .4 .4	.4 .4 .4 .4 .4 .4 .4 .4	.4 .4 .4 .4 .4 .4 .4 .4	.4 .4 .4 .4 .4 .4 .4 .4	.4 .4 .4 .4 .4 .4 .4 .4	.4 .4 .4 .4 .4 .4 .4 .4

Figure 1. Component level model of the Diesel Engine

Based on the likelihood DSM shown in Figure 1, the combined likelihoods of change propagation can be calculated on a system level. We used the algorithm described by Ariyo et al. (2007) to calculate numeric values (refer to the original article for a more detailed description). The algorithm first computes the combined likelihoods of changes propagating between two components across a system boundary. It multiplies the combined likelihoods between the initiating component and all the 'border' components within the same system (through which changes can propagate to the affected component in another system) with the respective likelihood of changes directly propagating from these 'border' components to the affected component. These values are aggregated across all possible paths between the two components, resulting in combined component-to-component likelihoods that take system boundaries into account. In the next step, the component-to-system likelihoods are calculated by aggregating the probabilities that a change will propagate from the initiating component to any component in the regarded system. Finally, these values are averaged across all the components in the initiating system to obtain combined system-to-system likelihoods. Figure 2 shows these values (rounded to two decimal places) in a system level model of the Diesel Engine.

	Combustion System	Transmission	Control and Electricals	Fuel System	Lubrication System	Starter System	Ignition System	Air Intake	Cooling System	Exhaust System
Combustion System		.22	.19	.26	.08	.10	.39	.03	.23	.20
Transmission	.57		.05	.19	.13	.14	.13		.04	
Control and Electricals	.42	.18		.69	.12	.72	.13	.39	.04	.28
Fuel System	.40	.06	.45		.11	.18	.05			
Lubrication System	.46	.11	.25	.09		.05	.05	.06		
Starter System	.24	.02	.08	.12	.02					
Ignition System	.49	.12	.05	.16	.02				.19	
Air Intake	.20		.10		.05					.20
Cooling System	.17	.02	.05				.09			
Exhaust System	.17							.03		

Figure 2. System level model of the Diesel Engine

5 EXPERIMENTAL RESULTS

To explore the implications of different granularities of the input product model on the process simulation, we conducted a set of simulation experiments. Due to the scope of this conference paper, this is not a fully comprehensive experimentation but rather aims to provide insights on some important points. The nature of the change propagation model and different granularities of input models have implications for the presented simulation model. Both the CPM and the algorithm to calculate change propagation likelihoods on different levels of granularity are static and analytical approaches. To simulate the dynamic design process based on product models with different levels of decomposition additional assumptions have to be made (see Section 5.3 for more details).

We choose the presented experiments firstly to represent the differences between the simulation results for the two input models without changes to the simulation model. Secondly, we analyse one possible way to account for the different granularity of input models, listing further options and noting that these will have to be included in a more exhaustive study. Table 2 gives an overview of the experiments conducted in the following sections and states the set-up of the main model parameters. The parameters that are manipulated with respect to the basic simulation of the original diesel engine DSM are shaded in the table. While the simulation allows for concurrency, the number of resources is limited to one. Even though this simplifies conditions significantly it also ensures that the effects of varying granularity can be investigated without influences from concurrency effects. The following sections sum up the observations in the experiments and reflect on the implications of input model granularity for the simulation model more generally.

Table 2. Overview of the simulation experiments

Pre-set parameters	5.1 (a)	5.1 (b)	5.2 (a)	5.2 (b)
Total simulation runs	10000	10000	10000	10000
Input product model	Fine	Abstract	Abstract	Abstract
Priority policy	0 – 24; 0	0 – 24; 0	0	0 – 24; 0
Number of resources	1	1	1	1
Probability of initiating change	0.1	0.1	0.1 - 0.32	0.3
Max. change propagation steps	5	5	5	5
Max. allowed maturity difference	2	2	2	2
Number of maturity levels (ML)	4	4	4	4

5.1 Observations with initial basic set-up

After calculating the likelihoods of change propagation on the system level (see Figure 2) several model parameters are adjusted in a basic way to enable an experimental comparison of the two models. In the first step, the durations to develop individual components are simply added up to derive a simplified estimate of the duration to develop the entire system they belong to. Because the original product model does not include durations we assume that all components take equal proportions of time to develop (in the absence of iterations). Consequently, the system takes this amount of time multiplied by the number of its components to develop. Initially, both the original and the aggregated model are simulated with the same basic parameters (Table 2). Figure 3 shows the performance of the tested policies (Table 1) relative to random task selection (policy 0) and a histogram of the total duration for 10,000 runs with policy 7.

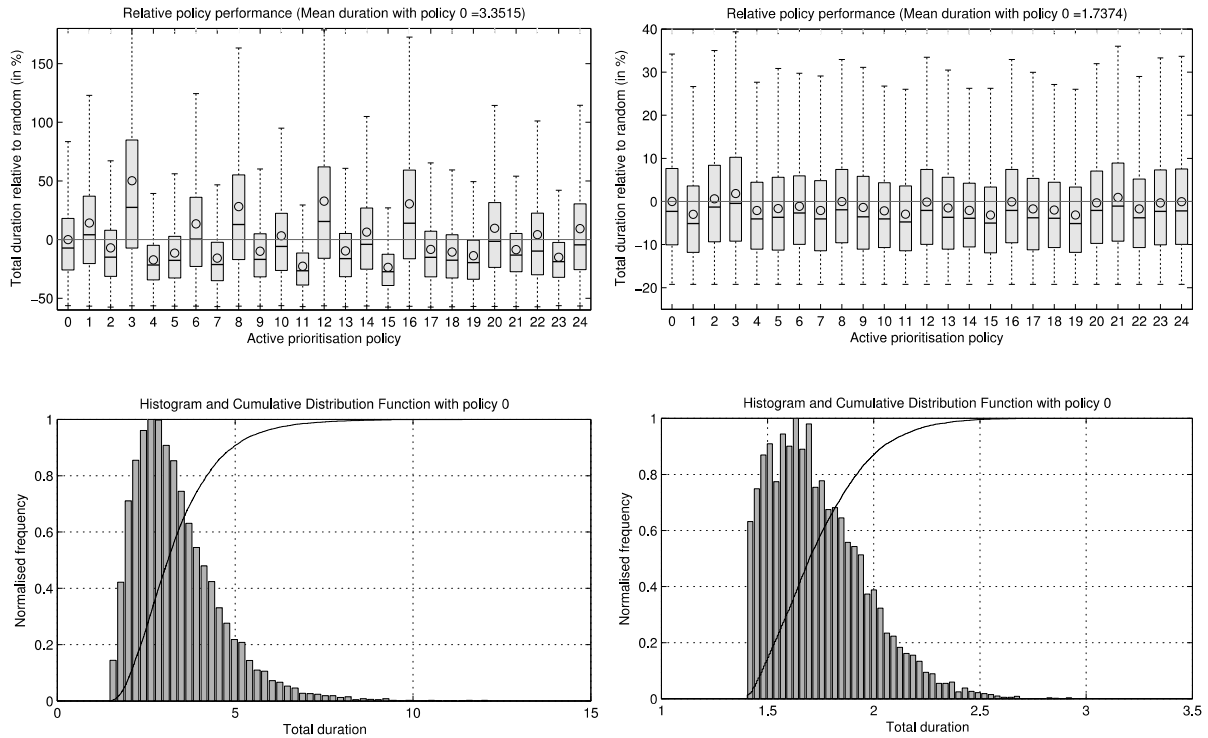


Figure 3. Simulation results for (a) original model (left) and (b) aggregated model (right)

The main differences between the original and the aggregated model immediately visible in Figure 3 are the less pronounced differences in relative policy performance and the decreased mean and variance in total duration. These effects are inter-related as the performance is measured in terms of duration. Although much less pronounced in the aggregated model, policy performance remains qualitatively similar (with the exception of the policies based on task duration - because of identical values in the original model only learning effects are accounted for). The lower duration of the aggregated model can be explained in several ways. Firstly, simply summing up the expected durations of components in a system underestimates the actual duration expected for the entire system as it omits iterations and rework due to intra-system interactions. Secondly, the identical change initiation likelihood for both simulations results in fewer such changes for the smaller input DSMs, given that changes are only initiated at the completion of task. Thirdly, related to the previous point, the fact that the number of maturity levels per component/system is identical leads to a lower number of total maturity steps for the entire product, which again decreases the number of changes. Fourthly, the impacts of change propagation may be underestimated as, due to lack of information and in absence of a mechanism to calculate them, they are assumed to be uniform across the aggregated DSM. The following Section investigates one way to account for the different properties of an aggregated model and Section 5.3 discusses the implications for simulation more generally.

5.2 Observations with adjusted experimental set-up

After having compared the simulation results of the original and aggregated model, some model assumptions are modified to account for changes in the level of granularity (see also Section 5.3). As discussed in the previous section, the main levers to adjust the simulation model are: likelihood of change initiation, aggregation of task durations on system level, number of maturity level per system and impacts of change propagation on system level. To analyse the effects of one of these parameters we perform a sensitivity analysis and compare the results to the simulation of the original diesel engine model.

We focus on the probability of change initiation, due to the limited scope of this study. Further possibilities to account for changing granularities have been outlined in Section 5.3 and will be briefly discussed but not analysed in detail in the next Section. To investigate the impact the change initiation likelihood has on the simulation results for the aggregated product model, a sensitivity analysis is carried out with the corresponding model factor. Figure 4 displays the results of the aggregated model with respect to the original model (displayed on the left side of the graph, see also Figure 3). This shows that the mean duration of the aggregated model under policy 0 and change initiation likelihood 0.3 corresponds to the original model. It has to be noted though that the characteristics of the distribution are different. For the original model (and the aggregated model with adjusted factor) the average is 3.35 (3.35), the median is 3.08 (3.17), the lower quartile 2.47 (2.62) and the upper quartile 3.93 (3.84). This shows that the results for the detailed model are more spread and more positively skewed (elongated tail at the right). Figure 5 displays the relative policy performance and histogram of policy 0 of the aggregated model with change initiation likelihood 0.3. Comparing this to results of the original model (Figure 3) shows that the impact of policies is less pronounced as well as a general decrease in variance as mentioned above. While the qualitative policy performance remains mostly similar, it is worth noting that the effects of unfavourable policies is underestimated.

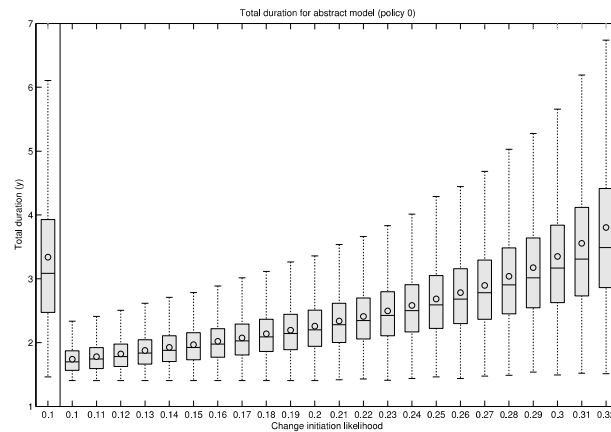


Figure 4. Sensitivity Analysis for change initiation likelihood with policy 0 (original model on the left side)

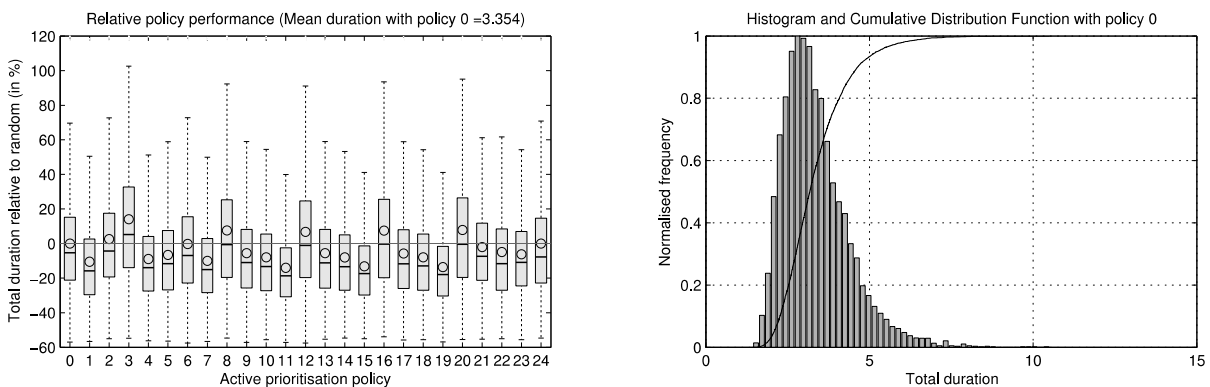


Figure 5. Simulation results for aggregated model with change initiation likelihood 0.3 (see Figure 3 (a) for a comparison with the original model)

5.3 Reflection on the implications of different granularities in the simulation model

In the simulation model used in this study, varying levels of input model granularity mainly has implications for the following model assumptions and variables: change initiation likelihood, task durations, impacts of change propagation, number of maturity levels, learning effects and priority rules. After having analysed the impacts of the first one in the previous section the remaining are briefly discussed in the following paragraphs.

The durations to complete parts of the design can be aggregated in several ways. One way is to sum up the durations to develop the individual components that are part of a system. While this is an easy solution it does, however, not account for system-internal iterations. It is also possible to simulate the development of the systems independently and then take the resulting durations as an input for a system level simulation. Other options include starting with one of the above and adding a factor to account for the amount of iteration and rework that is not accounted for in a more abstract model.

In the original simulation model, the change initiation likelihood is constant throughout the design process. This means that in an aggregated model fewer changes will be initiated due to the lower number of components. The change initiation likelihood value can thus be adjusted to account for these effects.

The algorithm by Ariyo et al. (2007) calculates the likelihood of changes propagating on different levels of granularity but does not account for their impact. The simulation model requires both values for its change propagation algorithm. So, either the impacts of change propagation have to be calculated in similar fashion or the impacts are assumed to be a certain value for the purpose of the simulation. Arguably, the impact of a change on the system level is likely to be bigger than on a component level. While this is partly accounted for by higher task durations the impacts of change propagation on different levels remains to be investigated.

In the original simulation model, components are assigned a discrete maturity level between 0 and 4. It can be argued that it makes sense to model maturity progression more finely on a system level, given that systems comprise several components with specific design maturity levels.

The learning effect used in the original model may be more appropriate on a component level. Although learning effects can also be expected on a system level their magnitude is expected to be lower due to the more distributed and fragmented nature of developing a bigger, more complex entity.

Lastly, while the priority rules themselves are generic to the design process they could be extended or adjusted in the light of different granularities. For instance, policies could be implemented that take into account how big a system is or how interconnected it is internally. It is also worth noting that the implications of these policies are different on a more abstract level, moving from operational planning and task scheduling to higher level strategic planning.

6 DISCUSSION AND CONCLUSIONS

In this study we have simulated design processes based on a product model with two different levels of granularity. This has shown that the simulation results are sensitive to shifts in the level of granularity, which highlights the importance of considerations regarding model granularity. In general, the simulation with the aggregated model exhibits less variance in the results and, in absence of adjustments, a reduced total duration. To obtain a similar total duration between the detailed and aggregated model we analysed the sensitivity to change initiation likelihood. It is possible to achieve identical mean total durations, which suggests that aggregated models can also be used for initial planning purposes. However, differences in the statistical distribution remain and the impacts of policies are underestimated, especially for unfavourable ones.

Due to the restrictions of this conference paper, the analysis had to be limited. We only consider two models of the same product on different levels of decomposition, which allows to derive initial insights but not a more general theory. Both the number of architectures and levels of granularity have to be increased and different decompositions included to attempt a comprehensive analysis. This study adopts a bottom-up algorithm to abstract from a detailed model. Further understanding of the role of model granularity could be gained by including a top-down perspective and investigate combinations of these two perspectives. For instance, a model could be detailed and abstracted again, analysing the effects and deriving guidelines to ensure consistency. Given that the decomposition and change probabilities depend on the perspective of the involved stakeholders, the influence of the original model on derivations with different levels of granularity should be addressed. Also, this article only

investigates the effects of one adjustment to the simulation model to account for varying input model granularity. In order to obtain a more exhaustive picture of the effects of such adjustments and their interactions, they have to be analysed in a simulation study with a range of input models. This will allow to quantify the relationship between model factors, input model properties and simulation results as well as determining valid combinations. Another limitation concerns the two approaches this study builds up on. Even though they are thought to have high face validity as they are synthesised from established and accepted concepts, they still only provide a particular perspective on the implications of model granularity. An empirical study would improve the understanding of the implications and handling of granularity decisions in industry and allow validating the results against real-world scenarios.

Model granularity is an important topic yet scarcely covered in the research literature. There is little discussion of how the granularity of a model was chosen and in many cases the assumption seems to be that a correct level emerged spontaneously. This study, along with the work of Chiriac et al. (2011), highlights the importance the level of granularity has for architectural analysis and design process planning. By extending the analytical change prediction method to a dynamic process simulation, we are able to draw conclusions on the effects of change propagation and rework on the design process. Being able to do this on different levels widens the scope of the model in terms of which planning decisions can be supported. Also, by directly linking system architecture with design processes the presented approach more generally enables investigating the impacts of model granularity across domains. We believe that the results obtained in this study, although currently limited in their scope, can be related to more general considerations about (simulation) model granularity and justify further research. For instance, the presented approach could be extended to study other cross-domain cases with an underlying network structure, such as information flow processes on different hierarchical levels in an organisation.

REFERENCES

- Ariyo, O.O., Keller, R., Eckert, C.M. and Clarkson, P.J. (2007), "Predicting change propagation on different levels of granularity: an algorithmic view", 16th International Conference on Engineering Design (ICED'07), Design Society, Paris.
- Browning, T.R. and Eppinger, S.D. (2002), "Modeling impacts of process architecture on cost and schedule risk in product development", IEEE Transactions on Engineering Management, Vol. 49 No. 4, pp. 428–442.
- Chiriac, N., Hölttä-Otto, K., Lysy, D. and Suk Suh, E. (2011), "Level of Modularity and Different Levels of System Granularity", Journal of Mechanical Design, Vol. 133 No. 10, p. 101007.
- Cho, S.H. and Eppinger, S.D. (2005), "A Simulation-Based Process Model for Managing Complex Design Projects", IEEE Transactions on Engineering Management, Vol. 52 No. 3, pp. 316–328.
- Clarkson, P.J., Simons, C. and Eckert, C. (2004), "Predicting Change Propagation in Complex Design", Journal of Mechanical Design, Vol. 126 No. 5, pp. 788–797.
- Jarratt, T.A.W., Eckert, C.M. and Clarkson, P.J. (2004), "Development of a Product Model to Support Engineering Change Management", Proceedings Fifth International Symposium on Tools and Methods of Competitive Engineering, Lausanne, pp. 331–342.
- Maier, J.F., Wynn, D.C., Biedermann, W., Lindemann, U. and Clarkson, P.J. (2014), "Simulating progressive iteration, rework and change propagation to prioritise design tasks", Research in Engineering Design, Vol. 25 No. 4, pp. 283–307.
- Oxford Dictionary (2014), available at:
http://www.oxforddictionaries.com/definition/american_english/granularity (accessed 3 December 2014).
- Pasqual, M.C. and de Weck, O.L. (2011), "Multilayer network model for analysis and management of change propagation", Research in Engineering Design, Vol. 23 No. 4, pp. 305–328.
- Simon, H.A. (1962), "The Architecture of Complexity", Proceedings of the American Philosophical Society, American Philosophical Society, Vol. 106 No. 6, pp. 467–482.
- Smith, R.P. and Eppinger, S.D. (1997), "A Predictive Model of Sequential Iteration in Engineering Design", Management Science, Vol. 43 No. 8, pp. 1104–1120.
- Wynn, D.C. (2007), Model-Based Approaches to Support Process Improvement in Complex Product Development, PhD thesis, University of Cambridge.
- Wynn, D.C., Eckert, C.M. and Clarkson, P.J. (2007), "Modelling iteration in engineering design", 16th International Conference on Engineering Design (ICED'07), Design Society, Paris.