INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITÉ DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

UNDERSTANDING DESIGN PROCESS ROBUSTNESS: A MODELLING APPROACH

Marek J. Chalupnik, David C. Wynn, Claudia M. Eckert and P. John Clarkson
University of Cambridge, UK

ABSTRACT

Mitigating the influence of uncertainty in design projects without sacrificing performance is challenging, but also important as it may reduce the risk of engineering projects delivering over time or budget. In this paper, we discuss process modelling and simulation as a route to understanding the impact of uncertainty and thereby understanding robustness in design processes.

We focus on robustness of process duration with regard to delays in constituent tasks. Firstly, we discuss some patterns by which processes respond to uncertainty by analysing how changes to the duration of an individual task can affect the overall duration of the process. Secondly, we highlight the complexity of process robustness analysis by using a discrete-event Monte-Carlo simulation to investigate the response of a 12-task mechanical component design process to uncertainty in task durations and rework. We argue that valuable insights can be gained by exploring the process response to modifying one input variable at a time. However, due to interdependencies between variables a multivariate approach would be necessary to fully investigate process robustness.

***Keywords**: Design process simulation, robust processes, uncertainty, Applied Signposting Model*

1. INTRODUCTION

Time and again, big engineering projects have failed to deliver on time and on budget (*e.g.*, [1]). One contributing factor is that, despite the advent of powerful information and knowledge management systems, uncertainty still prevails in project management. Mitigating the impact of uncertainty in engineering in general, and in New Product Development (NPD) in particular is a difficult task. Different approaches, such as Risk Management methods or Failure Modes and Effects Analysis (FMEA) have been proposed to achieve this goal. However, the focus of these approaches is mainly on the product. With the introduction of Taguchi methods (see *e.g.,* [2, 3]) much of the focus in the literature on handling uncertainty in engineering has been shifted towards the principles of Robust Design. Due to the lack of a precise definition of robustness in engineering design, the principles of Robust Design have been often used to denote very different meanings and encompass very different attributes, such as reliability or flexibility (*e.g.*, [4]).

Unlike the traditional focus of Robust Design in engineering, where analysis concentrates on the product, in this paper we investigate the robustness of the process, *i.e.*, the capability to deliver expected results in the presence of unexpected adverse factors. We assume that by managing design processes to improve their robustness the slippage rate of engineering projects could be reduced. The first step is to understand how process robustness may be investigated; this is the focus of this paper.

1.1. Paper overview

Following a brief discussion on product- and process-oriented approaches to robustness in section 2, a theoretical investigation of the effects of task duration on process outcome is presented in section 3. Section 4 discusses the complexity of robustness analysis and introduces the one-factor-at-a-time approach to analysing process robustness. In sections 5 and 6, based on modelling and simulating a simple mechanical component design process, we evaluate this approach and present implications for robustness analysis. In section 7, conclusions are given.

2. BACKGROUND: PRODUCT- VS PROCESS-ORIENTED ROBUSTNESS

The main aim in product-oriented approaches to robustness is to design a product with inherent robustness through incorporation of robust design principles in the design process. This may include techniques such as parameter setting for variance reduction or improving product robustness against noise. In product-oriented approaches, the performance of the design process is evaluated mainly through the robustness of the resulting product, and from the wider perspective, through its capability to reduce quality loss to society (Figure 1a). In contrast, our process-oriented approach focuses on the performance of the design process itself. The quality, reliability, cost, performance robustness and other attributes of the product are assumed to be outside the scope of process robustness analyses (Figure 1b).

In measuring the performance of the process, several criteria, for example the lead-time or development cost, can be used. Process robustness can therefore be defined in terms of alternative measures of process performance, and different viewpoints for its investigation are possible. We discuss this in more depth in [5]. In this paper, we focus on one form of analysis: examining the robustness of process duration with respect to individual task durations. This focus was chosen because understanding the ultimate impact of task delays on the process is widely perceived as a challenge in industry.

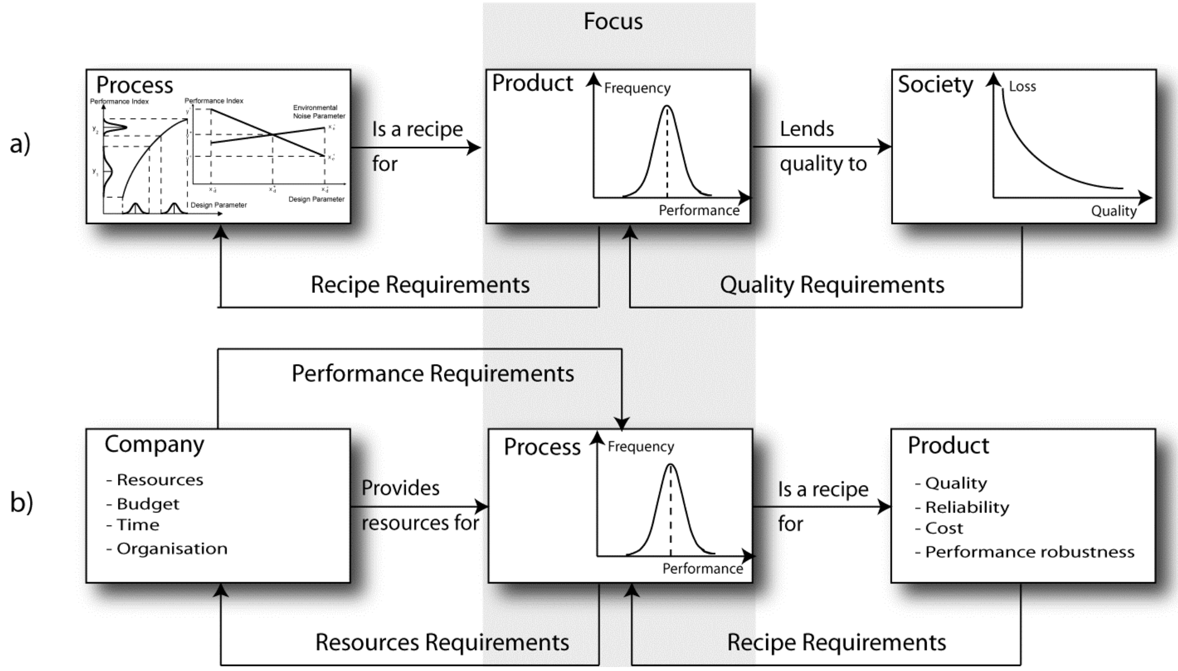


Figure 1. A comparison between product- and process-oriented approaches to robustness: a) product-oriented approach, b) process-oriented approach

3. EFFECTS OF TASK DURATION ON PROCESS OUTCOME

The way process duration is affected by delayed tasks is dependent on the structure of information flows as well as the duration of the delay. Short delays are often unlikely to cause a process slippage regardless of the type of the task and its position on the critical path. Conversely, long delays may cause disruption of the entire project even if the task is not originally on the critical path.

To facilitate discussion about process robustness with regard to duration, the following section explains several possible mechanisms by which a delay in completion of a single task influences total process duration and the execution order of subsequent tasks. Although such delays can originally be caused by a number of causes such as too optimistic estimates or subcontractor's late delivery, their ultimate effect is determined by the process structure (*i.e.*, information flows), resource constraints and task selection policies. In this context, task selection policies govern which task is attempted when more than one task is possible to start but they cannot be executed in parallel due to resource limitations.

3.1. Structure absorbs delay

For a process structure to absorb a task delay, *i.e.,* to ensure that total process duration remains unchanged or at least does not exceed the maximum permitted value, several conditions have to be met. Most importantly, the task should not be on the critical path of the project, which means that there are no downstream tasks whose execution immediately waits on the release of the task output. To illustrate this point, consider Figure 2 in which tasks *1*, *3* and *4* are on the critical path and the deferral of any one will delay the whole process. In contrast, task *2* is an example of an activity that is not on the critical path (for more discussion on critical path analysis in design see *e.g.*, [6]).

Among other conditions that have to be met for process duration to remain unchanged following task delay is that the overdue task should not cause any risk, cost or quality perturbations that may eventually result in process slippages. These secondary interactions are not considered in our analysis.

3.2. Structure propagates delay

An overdue task can postpone the completion of a certain milestone or even the entire process. The ultimate effect of a task delay on the project or milestones depends upon several factors, including: the absorption capacity of buffers in the process; policies regarding process execution and resource usage; and the structure of the activity network itself. The following subsections describe two different effects that a task delay of the same magnitude can have on the process depending on its structure.

Propagated delays

Once a task is on the critical path its delay will propagate through the process. For example, if subsequent tasks are executed according to Figure 2, the propagated delay will reach the end of the task chain, whose length depends on the number and capacity of buffers in the process.

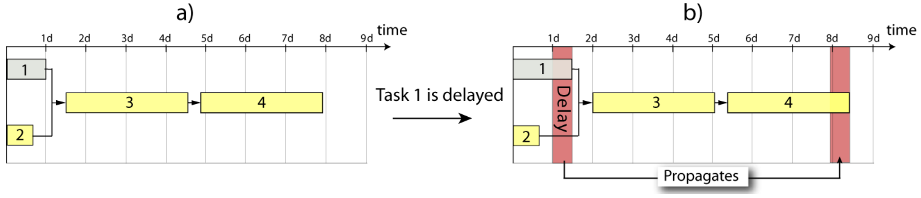


Figure 2. An illustration of a task on the critical path

Propagated delays with resource inefficiency

Rescheduling one task can cause inefficiency in resource usage in addition to delays in subsequent tasks. As illustrated in Figure 3, an initially continuous profile of resource usage can become disjoint following a task delay. The resulting discontinuities represent resource inefficiency. This example shows that simple task rescheduling can have bearing not only on a timely accomplishment of the project but also on other common measures of process performance.

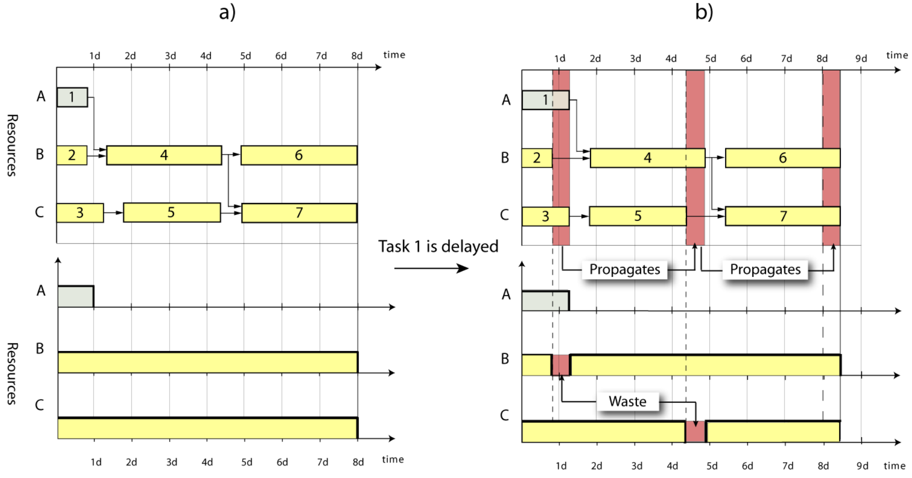


Figure 3. Propagation of a task delay across the project (top row) and a resource usage profile illustrating inefficiencies due to the delay (bottom row): a) original project b) project with a task delay. A, B and C represent different types of resources.

Propagated delays may be commonly observed in industrial practice and are intuitive to understand in principle. In their simplest form, the possibility of such delays can be recognised and mitigated in a plan (*e.g.*, using buffers) without requiring sophisticated analysis. However, mitigating uncertainty in plans can become significantly more complicated if additional dimensions of process performance are taken into account and the process plan is optimised from several points of view.

3.3. Structure accumulates delay

The concept of accumulated delays is similar to propagated delays, as they both relate to the impact of a delay in one activity upon tasks that are downstream in the process. In accumulated delays, a delay is magnified as it propagates through the process. This is often caused by rework. How much process execution will be behind schedule depends on the prevalence and particular characteristics of the reworked tasks. Even if rework does not take place, process or milestone completion may be delayed since the initial delay will propagate to the subsequent tasks that are not in the rework cycle (Figure 4c).

This type of slippage may be difficult to distinguish from propagated delays during actual processes. However, analysing them separately is useful for at least two reasons. First, this helps explain some apparent discrepancies between process plans and observed process performance. Second, the analysis can help identify tasks whose failure or postponement may most seriously affect the outcomes of processes with significant rework. Such processes are common in engineering design.

In the previous discussion, task delays were assumed to follow from either rescheduling or extended duration. In accumulated delays, rescheduling is not considered. This is because a case in which the first execution of the task is rescheduled to a later date does not necessarily result in the accumulation of delay, since the execution of reworked tasks may be on schedule and the initial postponement will simply propagate without any increment.

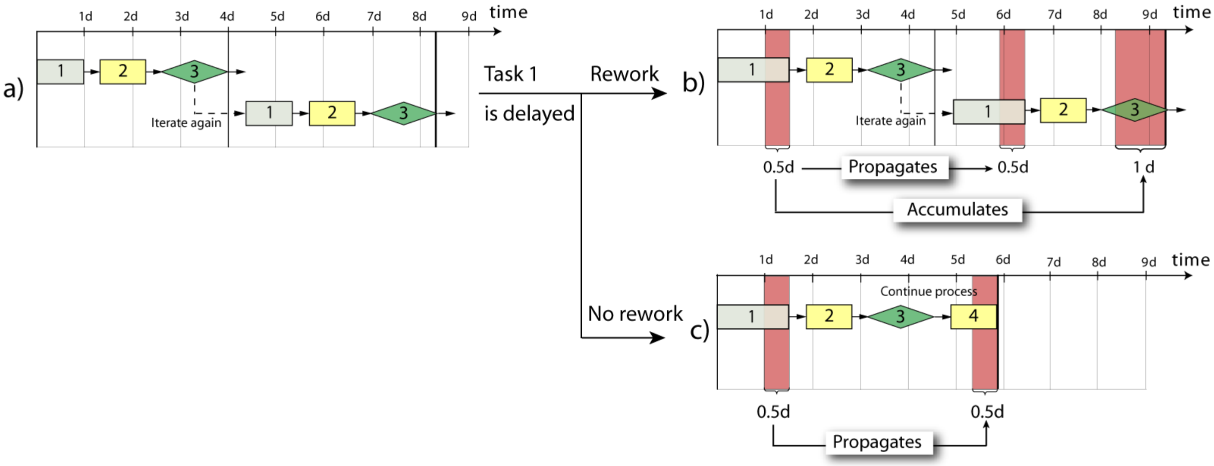


Figure 4. Accumulation of delays due to rework

3.4. Structure causes process duration to reduce following delay

As already demonstrated in the case of propagated delays, consideration of resource constraints reveals a significantly more complex problem. Resource constraints can cause counter-intuitive behaviour and trade-offs in process performance. Figure 5 illustrates a case in which resource constraints cause an *improvement* in process performance following a delay in one task. In this example, a policy regarding resource utilisation (continuous utilisation of resource *B*) and/or a policy concerning the priority of activities in the project (task *3* after or before task *4*) contribute to the outcome.

Such counter-intuitive response to delays illuminates several issues important in planning design processes. First, in order to fully assess the effectiveness of the policies in use regarding task selection and resource utilisation, possible task delays should be investigated. For example, a *longer-task-first* or *riskier-task-first* task selection policy may prove inefficient if patterns of criticality of tasks in the process change due to rescheduling or changes in initial estimates of duration.

Second, resource utilisation efficiency from a local perspective may not guarantee global efficiency in resource usage. To illustrate this, consider the resource profile depicted in Figure 5a, where a policy favouring efficient and continuous utilisation of resource *B* leads to inefficiency in total resource use and, consequently, to the longer duration of the process.

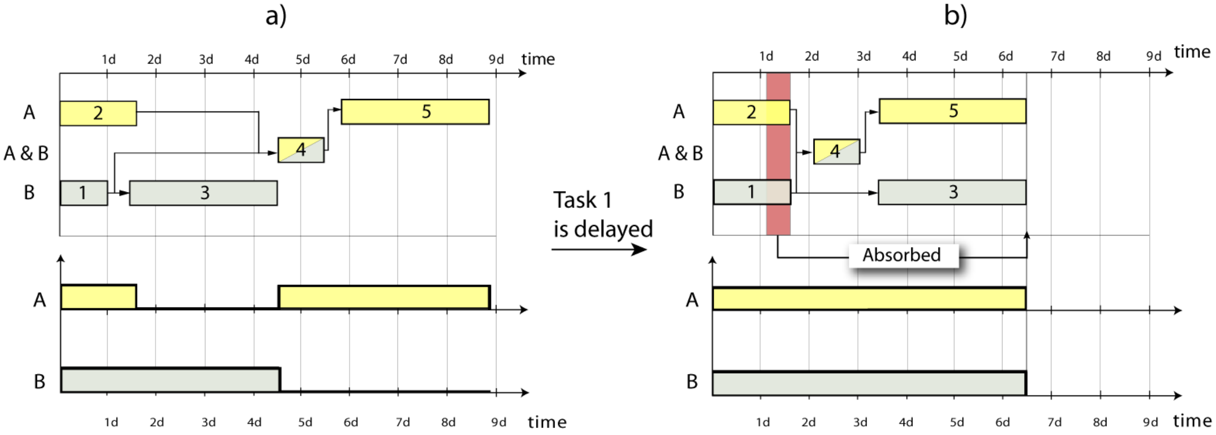


Figure 5. Counter-intuitive process behaviour due to a task delay (top row) and a resource usage profile illustrating more efficient usage due to the delay (bottom row): a) original project b) project with a task delay. A and B represent different resources.

3.5. Summary

The previous sections have illustrated four responses of process duration to delays in individual tasks, according to the structure of information flows in the process: *absorption*, *propagation*, *accumulation* and *negative response* to these delays. This shows that interactions between single task durations and process outcomes can be complex. Additionally, the effects of these interactions may be non-linear since structural configurations may interact to amplify or buffer one another's response. To summarise, it is difficult to evaluate the sensitivity of the process to even a single task delay without detailed analysis. Identifying ways to reduce this sensitivity is the essence of robustness analysis.

4. COMPLEXITY OF ROBUSTNESS ANALYSIS

Due to various sources of uncertainty in the design process (see for example [7]), several aspects of process robustness can be analysed. However, capturing, representing and distinguishing uncertainty in engineering design models is difficult. In this section, we look at some of those challenges.

4.1. Challenges

Process uncertainty can manifest in a number of ways. However, representing uncertainty in a model is restricted by the modelling framework used. For example, some process behaviour may not be possible to model within a given framework due to the process description, algorithms or statistical methods used. This may reduce the scope of robustness analysis.

Notwithstanding the limitations of a modelling framework, the very nature of design process models also restricts robustness analysis for two main reasons. First, models are always simplifications of reality and modellers, who can never fully capture the complexity of uncertainty inherent in design, must subjectively identify the most important factors to be included in any analysis. Second, the complexity and size of design process models can limit the choice and number of factors included in robustness analysis due to its computational expense. For example, identification of factors that may contribute to project failure is extremely difficult once the modeller can control hundreds of stochastic variables that are interdependent in a potentially complex way. Even though simulation may facilitate this task by helping the modeller to explore probabilistic aspects of process behaviour, analysing the relationships between numerous variables is computationally expensive and in many cases practically infeasible.

Uncertainty stemming from the model itself is another important limitation in process robustness analysis. While evaluating the influence of uncertainty on the process, the modeller has to take into account the structure of the model and the assumptions made in order to draw reliable conclusions with regard to process robustness [8].

4.2. One-factor-at-a-time analysis of process duration

There are numerous possible sources of change in process duration. For example, uncertain task outcomes and durations, variations in product requirements and available development resources all can impinge on project cost, efficiency and its completion date. For the reasons discussed above, in all but the simplest cases it is often not feasible to capture all these factors simultaneously within a model. In the remainder of this paper, we therefore restrict our focus to investigating the robustness of process duration against undesirable changes (*i.e.*, increases) in task durations. We discuss a one-factor-at-a-time approach to robustness analysis, *i.e.*, only the duration of one task at a time is assumed to change compared to the baseline configuration of the process. This approach readily lends itself to extension. For example, we show in section 5 how evaluation of the impact of each task duration on process duration can be extended to include variations in rework likelihood and resource constraints.

5. A LABORATORY EXPERIMENT

In this section, we study a simple mechanical component design process to examine the usefulness of a one-factor-at-a-time analysis of process duration, and to show how this approach may be used to evaluate process robustness.

Many mechanical design processes can be described as: "*the definition of a geometry to carry a given set of loads subject to constraints on the allowable bulk stress within the component and on local stress concentrations"* [9]. We use the 12-task model of such a process described by Clarkson *et al.* [9] as the basis for this experiment. The model has been re-constructed using P3 Signposting, a design process modelling tool which provides the simulation functionality utilised in the subsequent analysis [10].

5.1. The Applied Signposting Model (ASM)

The P3 Signposting software implements a task-based process modelling framework called the Applied Signposting Model (ASM) [11]. The ASM incorporates process simulation based on the following assumptions:

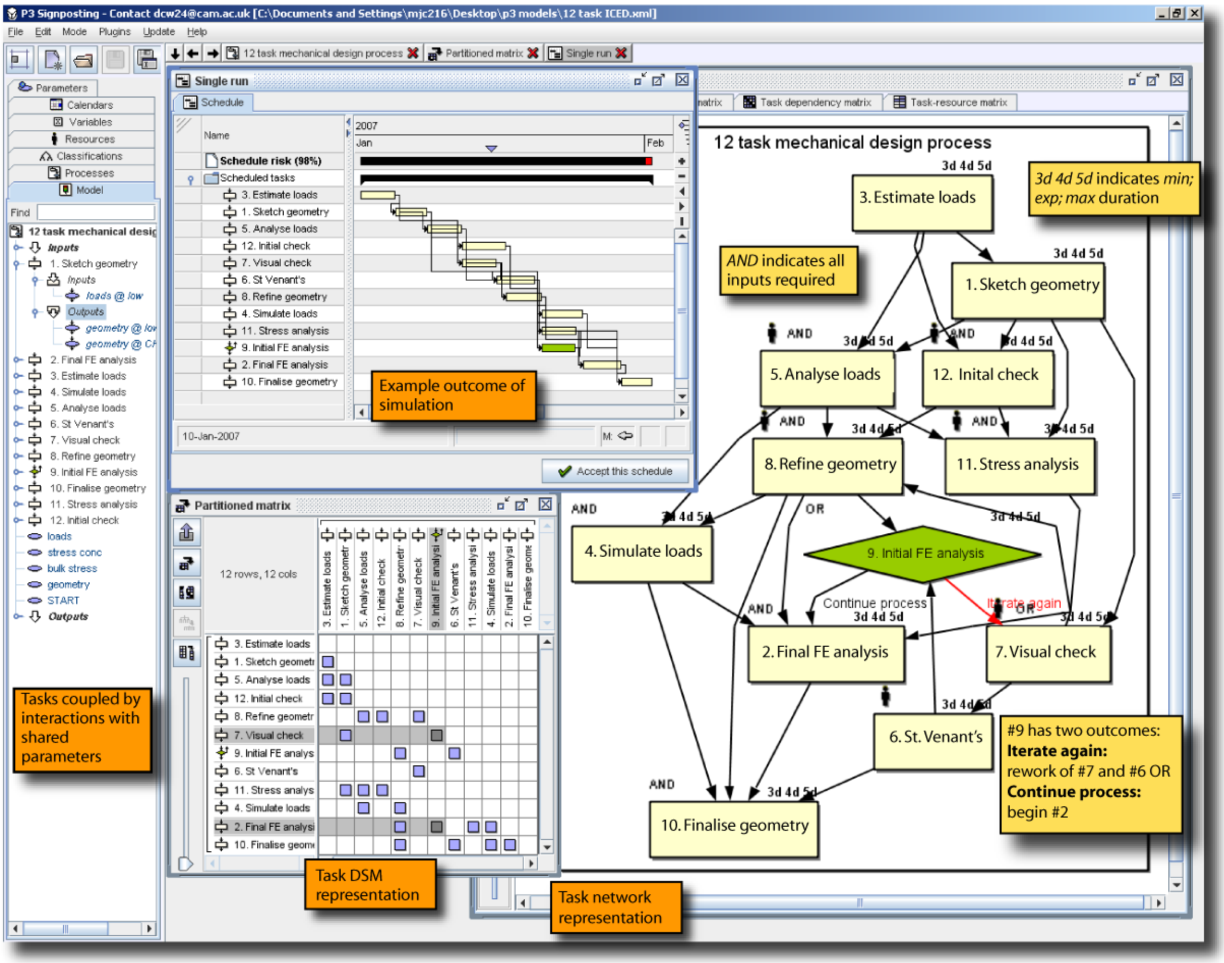
- tasks are available to begin only when their input information is available;
- tasks must be re-attempted when input information changes (*i.e.,* following iteration);
- tasks do not affect process behaviour during execution, *e.g.*, by releasing information prior to completion.

The approach uses an extended Petri-net approach to model information flows between tasks in a design process. This forms the basis of a Monte-Carlo discrete-event simulation. Full detail of the ASM modelling approach is available in [11].

5.2. 12-task model of a mechanical design process

The model used for analysis is composed from the following tasks: *1. Sketch geometry, 8. Refine geometry, 10. Finalise geometry, 3. Estimate loads, 5. Analyse loads, 4. Simulate loads, 7. Visual check, 6. St. Venant's, 9. Initial Finite Element (FE) analysis, 12. Initial check, 11. Stress analysis, 2. Final FE analysis* [9]. Task Design Structure Matrix (DSM) and task network representations of the model are shown in Figure 6.

The duration of each task in the simulation model is characterised using a triangular probability density function. A single rework cycle is included in the model. The number of iterations is determined by the probability of rework on each attempt, defined as an attribute of *9. Initial FE analysis*. Each time rework is required tasks *6*, *7* and *9* must be re-attempted. It is assumed that the probability of rework and the task duration probability density functions remain constant, irrespective of the number of iterations which are attempted (*i.e.,* no learning effects). In the baseline configuration, there are no resource constraints which might limit concurrency. Figure 6 shows one possible order of task execution and illustrates that certain independent tasks may be conducted in parallel.



Tasks coupled by interactions with shared parameters

Example outcome of simulation

Task DSM representation

Task network representation

3d 4d 5d indicates min; exp; max duration

AND indicates all inputs required

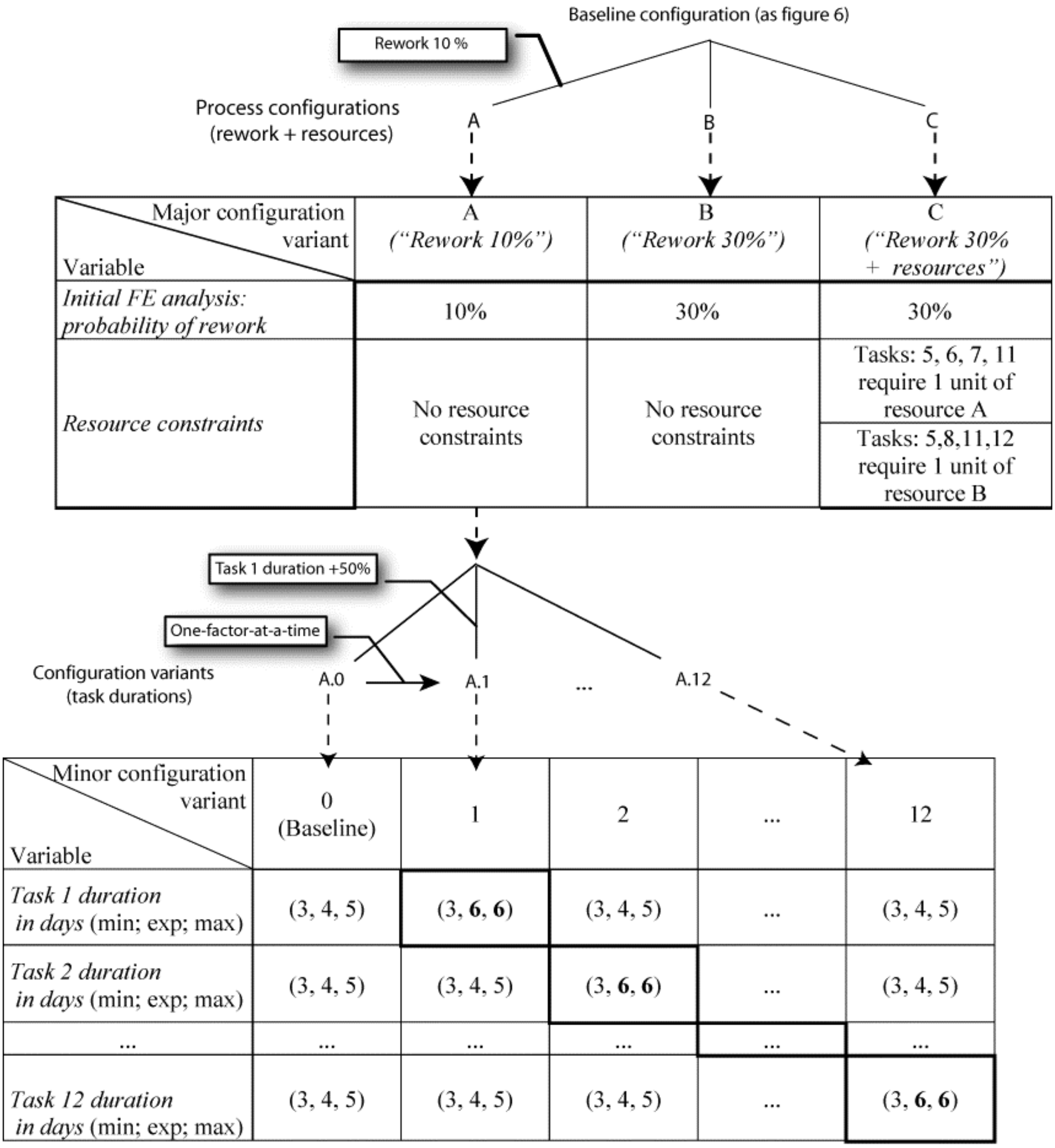
#9 has two outcomes: **Iterate again:** rework of #7 and #6 OR **Continue process:** begin #2

Figure 6.Task DSM, task network (i.e., PERT chart) and example Gantt chart views of the 12-task design process modelled using P3 Signposting [10]

5.3. Method description

To analyse the process a number of variants were constructed by modifying the baseline configuration (Figure 7). In each such variant, a single change of the same type (an increase in the expected and maximum duration of a specific task) and the same relative magnitude (accordingly 50% and 20%) was introduced and 10 000 simulations were run. This proved sufficient to identify and exclude from analysis purely random effects in process behaviour (see convergence plot in Figure 8). In each variant, attributes of the remaining eleven tasks were unchanged from the baseline model (Figure 7). This allowed investigation of the sensitivity of process duration with respect to changes in individual tasks' durations. Such analysis is useful for a number of purposes, because changes in task duration can result from several sources of uncertainty in a process (*e.g.*, changes in requirements, delays in subcontractor delivery *etc.*).

These 12 'minor' variants (plus baseline) were combined with 3 'major' configuration changes in each set of simulations, such that every minor variant was evaluated in the context of a major configuration change (Figure 7). This accounted for other sources of uncertainty in the process and permitted assessment of the one-factor-at-a-time approach in different process configurations. Two of these major changes were different levels of the probability of rework (controlled by *9. Initial FE analysis: probability of rework* variable). The third was designed to illustrate the effect of resource constraints on robustness analysis (configuration *C* in Figure 7). In this configuration, resource requirements were therefore added to prevent parallel execution of the two groups of tasks: *5*, *7* and *12* and *6*, *8* and *11* (see Figure 6). This allowed evaluation of task criticality which is dependent on resource constraints, as discussed in previous sections.



Major configuration variant / Variable	A *(“Rework 10%”)*	B *(“Rework 30%”)*	C *(“Rework 30% + resources”)*
Initial FE analysis: probability of rework	10%	30%	30%
Resource constraints	No resource constraints	No resource constraints	Tasks: 5, 6, 7, 11 require 1 unit of resource A Tasks: 5,8,11,12 require 1 unit of resource B

Minor configuration variant / Variable	0 (Baseline)	1	2	...	12
Task 1 duration in days (min; exp; max)	(3, 4, 5)	(3, **6**, **6**)	(3, 4, 5)	...	(3, 4, 5)
Task 2 duration in days (min; exp; max)	(3, 4, 5)	(3, 4, 5)	(3, **6**, **6**)	...	(3, 4, 5)
...	...	...	...	...	...
Task 12 duration in days (min; exp; max)	(3, 4, 5)	(3, 4, 5)	(3, 4, 5)	...	(3, **6**, **6**)

Figure 7. Analysis proceeds by simulating alternative configurations of the same process. Each configuration comprises a major component (the % likelihood of task #9 revealing rework) and a minor component (the durations of all tasks 1-12). All tasks have equal duration in the baseline process; minor variants 1-12 extend the duration of tasks 1-12 respectively. Refer to Figure 6 for process structure.

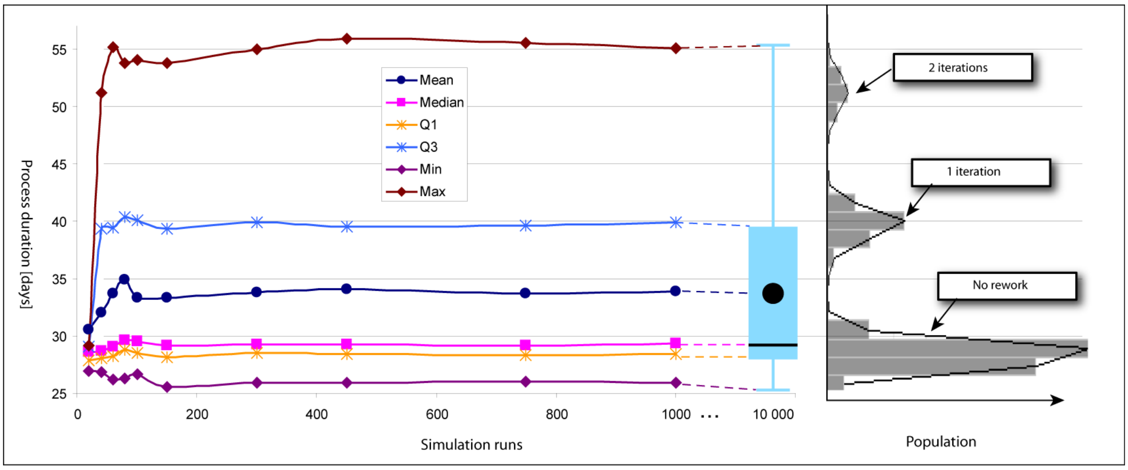


Figure 8. Convergence of process duration indicators (left) and simulation results showing multimodal distribution (right) for process configuration B.0 ("30% rework"). Figure indicates that the indicators plotted in Figure 9 converge well before the 10 000 simulation runs used. Results are similar for the other configurations.

5.4. Simulation results

Figure 9 summarises simulation results for each process configuration. The process duration distributions for each major + minor configuration variant are represented as box plots and ordered according to the values of their third quartiles. The boxes show 50% instances between the first and third quartiles, the whiskers indicate the minimum and maximum regular values and the dots show the statistical means.

Many statistical measures of process performance can be calculated from the simulation results. The choice of criterion for this analysis has been informed by behavioural characteristics of the model; in particular, its predicted duration has multi-modal distribution which decays in an exponential manner due to the unconditional likelihood of rework associated with task *9* (Figure 8). The third quartile value of this distribution has been found to adequately indicate changes in the distribution.

In each major configuration, three main categories of minor variants can be identified by inspection. *Group 1* consists of those configurations whose respective distributions are similar to the original process. These correspond to the tasks to which the baseline process is most robust (Figure 9, left-most bars). *Group 2* includes those variants that are in the centre of the spectrum, and which represent distributions moderately shifted towards higher values of process duration. *Group 3* is composed of the variants that correspond to greatest change in process duration (Figure 9, right-most bars); delays in the corresponding tasks have greatest potential to affect total duration.

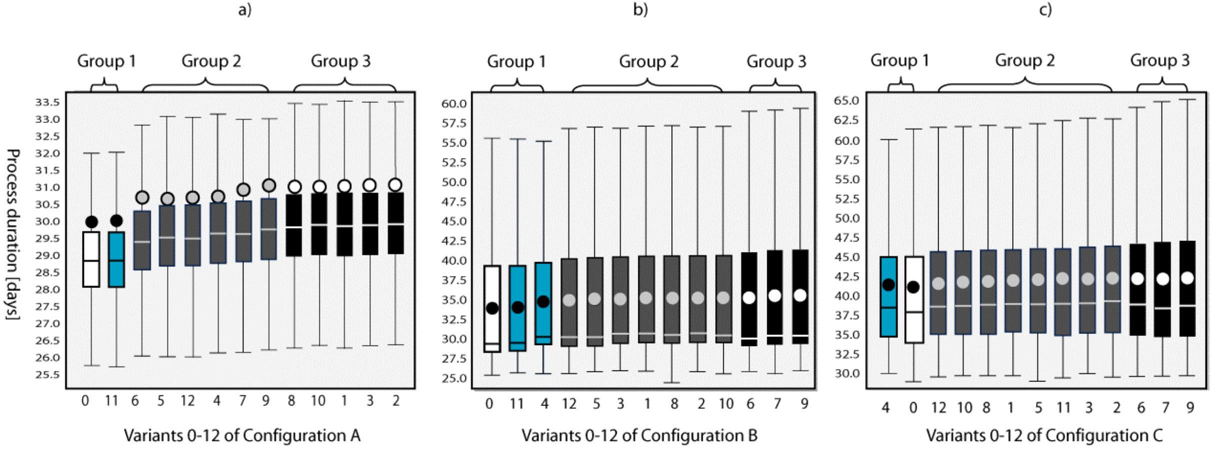


Figure 9. Simulation results showing total duration for process configurations: a) A ("10 % rework") b) B ("30% rework") and c) C ("30% rework + resources"). Different groups of the boxes represent different process sensitivity. Refer to Figure 7 for configuration details.

Categorisation of the variants in this way serves two purposes. First, it allows assessment of the sensitivity of process duration to changes in a given task duration, as the group a variant belongs to provides an indication as to the criticality of the task which was perturbed. Second, it facilitates comparison with the baseline configuration. For example, such comparison can reveal counter-intuitive process behaviour as configurations for which the duration distribution is to the left of the baseline process indicate faster project completion *following an increase in duration of one constituent task* (task *4* in Figure 9c). Although such counter-intuitive behaviour could be caused by statistical variability in the simulation, this has been ruled out by the convergence analysis. In this case, the counter-intuitive response is a complex effect of the process structure as discussed in section 3. This could be caused by erroneous assumptions during modelling, or could reflect the actual behaviour. This difficulty of attribution highlights the importance of a theoretical understanding of the simulation model in order to interpret its results.

Changes due to the major configurations of the model are reflected in different orderings of the minor variants on charts 9a-c. This indicates changing relative importance of the tasks on process duration.

5.5. Contingent task criticality

Due to characteristics of the information flow in the analysed process (Figure 6), there is only one task (*11. Stress analysis*) that never (under simulated conditions of 50% and 100% change of task duration) appears on the critical path. Regardless of the amount of rework in the process, changes in the task duration do not lead to any delays in process execution. This indicates that the project is robust against changes in the duration of this task, for the configurations examined.

Apart from *11. Stress analysis*, all the remaining tasks are potentially critical. They can be classified into three groups:

- Tasks that will always be on the critical path, regardless of the amount of rework in the process and the task execution policy used: *3. Estimate loads, 1. Sketch geometry, 10. Finalise geometry, 8. Refine geometry* and *2. Final FE analysis*. These tasks are always on the critical path. Any delays in their completion will lead to process rescheduling. In the case of low probability of rework (10%), these tasks are most critical to timely completion of the overall project (Figure 9a). Accordingly, the relative value of one-dimensional measures of project robustness against changes in the durations of these tasks is low.
- Tasks that can become critical once their completion is delayed and/or an unfavourable policy regarding task execution order is exercised: *5. Analyse loads, 4. Simulate loads and 12. Initial check*. As the tasks occupy the centre of the chart, the process is moderately robust against changes in their durations. This is especially true for *4. Simulate loads* whose criticality at the higher probability of rework is significantly reduced (Figure 9b).
- Tasks whose criticality is mainly dependent upon the amount of rework: *6. St. Venant's*, *7. Visual check* and *9. Initial FE analysis*. In these cases, the higher the probability of rework, the more critical the tasks become (a rightwards shift in Figure 9b compared with Figure 9a). An initial change in task duration will be magnified according to the accumulation mechanism discussed in section 3.3. The value of one-dimensional measures of project robustness against changes in the durations in these tasks is low and decreases with increasing probability of rework.

6. FINDINGS AND IMPLICATIONS FOR PROCESS ROBUSTNESS ANALYSIS

The analysis and experiment have provided five insights into robustness analysis. These are described below.

6.1. Importance of understanding causal factors in applying simulation analysis

Due to the complex interplay between propagation, accumulation and absorption of the impact of increased task duration on process performance, evaluating the sensitivity of even a simple 12-task process to a single task delay is very difficult. Simulation can greatly facilitate this task by providing insights into dependencies between process structure and behaviour. However, designing, analysing and interpreting the results of simulation requires an understanding of causal factors that contribute to robustness. Further theoretical investigation is thus needed to identify those factors.

6.2. Contingent process robustness

Among the most important implications stemming from the analysis of the simulation results is the observation that process sensitivity to changes in task duration, due to its relationships with other process characteristics such as rework or a policy regarding task execution order, is not an absolute measure of task criticality. This means that process robustness against changes in task duration is contingent upon other sources of uncertainty in the process. This in turn indicates the multi-faceted nature of the robustness concept and highlight the limitations of one-factor-at-a-time analysis.

6.3. Counter-intuitive behaviour

Another important, and perhaps the most interesting observation is that under the resource constraints (Figure 7) the analysed process exhibits what can be described as negative robustness: a 50% increase in the duration of *4. Simulate loads* leads to a decrease in process duration (measured by the value of its third quartile). Although one possible explanation of this counter-intuitive behaviour was given in section 3.4, further work will be needed to investigate other possible causes.

6.4. Limitations of one-factor-at-a-time analysis

The above analysis was restricted to examining the robustness of process duration against changes in individual task durations only. This cannot account for combinations of factors that may have non-linear effects. For example, identification of critical tasks in a process using one-factor-at-a-time analysis is not possible if delays in several tasks occur. This is because such delays, in addition to being propagated, accumulated or absorbed by the process structure according to the mechanisms described in this paper, interact with each other and might cancel out each other's effects. Further research based on a multivariate approach is therefore needed to investigate the relative importance of such effects and develop computationally feasible analyses to explore them.

6.5. Other implications for process robustness improvement

In order to rank the causal factors that contribute to process robustness and formulate recommendations to improve robustness, one has to have a good understanding of the assumptions made during modelling and simulation and of the limitations of the approach that has been used. This is because, firstly, the main causal factors that have been identified via simulation in one process model (for example, in this paper, the likelihood of rework) might not be relevant to other processes or modelling assumptions. Secondly, due to sources of uncertainty not included in the analysis, or insufficient depth of analysis between considered sources, the relative importance of previously identified causal factors may be radically changed. Consequently, research into the sensitivity of analysis results to changes in modelling and simulation assumptions will be needed to develop a systematic approach to process robustness improvement.

7. CONCLUSIONS

Uncertainty in engineering design can manifest itself in a number of ways. Hence, analysing the robustness of even simple design processes can be very complex. By using a discrete-event Monte-Carlo simulation to investigate the response of a 12-task mechanical component design process to uncertainty in task durations and rework, we highlighted the complexity of process robustness analysis and difficulties in identifying and ranking the causal factors that contribute to process robustness. Two general conclusions were drawn from this experiment:

- Understanding the origin of complex and sometimes counter-intuitive process behaviours is necessary to design appropriate simulation experiments and interpret their results.
- Task criticality cannot be fully investigated by exploring process response to changes in task duration alone, since other sources of uncertainty such as rework likelihoods and policies regarding task execution order also play an important role.

The next steps in our research will be to investigate process robustness against changes in resource availability and to look at those scenarios where a combination of factors contributes to a project failure. Since such analysis is computationally expensive, one possible research avenue will be to use more advanced statistical methods *e.g.*, those used in designed experiments. In the longer run, we plan to extend our analysis to include trade-offs between measures of process performance and to verify our approach through the investigation of more complex design processes.

REFERENCES

[1] Matta, N.F. and Ashkenas, R.N. Why good projects fail anyway. *Harvard Business Review, September*, 2003, pp. 109-114.

[2] Taguchi, G. and Clausing, D. Robust quality. *Harvard Business Review, January-February*, 1990, 65-75.

[3] Phadke, M.S. *Quality engineering using robust design*. (Prentice Hall, Englewood Cliffs, N.J., 1989).

[4] McManus, H.L. and Hastings, D.E. A framework for understanding uncertainty and its mitigation and exploitation in complex systems. In *Fifteenth Annual International Symposium of the INCOS, Rochester, NY*, 2005.

[5] Chalupnik, M.J. Modelling design processes to improve robustness. In *6th Integrated Product Development Workshop, IPD 2006,* Schonebeck/Bad Salzelmen b. Magdeburg, Germany, 2006.

[6] O'Donovan, B., Eckert, C.M. and Clarkson, P.J. Simulating design processes to assist design process planning. In *DETC'04, ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference* Salt Lake City, Utah, USA, 2004.

[7] Browning, T.R. Sources of schedule risk in complex system development. *Systems Engineering*, 1999, 2(3), 129-142.

[8] Choi, H.J. A robust design method for model and propagated uncertainty. (PhD thesis, Georgia Institute of Technology, 2005).

[9] Clarkson, P.J., Melo, A.F. and Eckert, C.M. Visualization of routes in design process planning. In *International Conference on Information Visualisation (IV2000), University of London, UK*, 2000, 155-164.

[10] Wynn, D.C., Nair, S., Bell, C.P., Chalupnik, M.J. and Clarkson, P.J. P3 Signposting Users' Guide. *Draft document, University of Cambridge, 2006 (http://www-edc.eng.cam.ac.uk/p3).*

[11] Wynn, D.C., Eckert, C.M. and Clarkson, P.J. Applied signposting: a modeling framework to support design process improvement. In *Proceedings of ASME IDETC/CIE 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, (DETC2006-99402), 2006.

Contact: Marek J. Chalupnik
Engineering Design Centre
University of Cambridge
Trumpington Street
Cambridge CB2 1PZ
UK
Phone: +44 – 1223 – 332758
Fax: +44 – 1223 – 766963
E-mail: mjc216@cam.ac.uk
URL: http://www-edc.eng.cam.ac.uk/people/mjc216.html