INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008
Dubrovnik - Croatia, May 19 - 22, 2008.

DESIGN 2008

ANALYSING THE RELATIONSHIP BETWEEN DESIGN PROCESS COMPOSITION AND ROBUSTNESS TO TASK DELAYS

M. J. Chalupnik, D. C. Wynn, C. M. Eckert and P. J. Clarkson

Keywords: design process robustness, robustness profile, hypothetical fragments, process simulation, Applied Signposting Model (ASM)

1. Introduction

Planning and executing New Product Development (NPD) projects is a challenging task. This stems in part from uncertainty in their behaviour – for instance, in the duration and outcome of their constituent activities or in the availability of resources to perform these activities. This paper is motivated by the hypothesis that engineering design processes can be configured to be robust, i.e. to deliver acceptable outcomes despite uncertainty, and that this could help project managers to deliver their projects on time and on budget.

A process' architecture plays an important role in determining its behaviour (Whitney et al. 2004). Several authors have discussed how design process architectures could be modified to better deal with uncertainty and hence to be more robust. This existing research falls into two categories. In the first category, authors focus on providing explanatory frameworks and categorisations of the ways in which system robustness can be analysed and improved. For instance, McManus and Hastings (2005) propose a framework which illustrates the connection between system architecture and robustness. However, their framework does not focus on the design process in particular and it is not clear how it could be applied to improve design processes in practice. Chalupnik et al. (2008) attempt to address the first limitation by discussing the robustness of a design process in terms of its response to delays in individual tasks. They highlight four ways in which a delay in completion of one task can influence total process duration: propagation, accumulation, absorption and reduction (Table 1). They illustrate how these effects can arise from certain arrangements of small numbers of tasks and resource constraints. Although this classification can help to explain process robustness, Chalupnik et al. (2008) only show that each of the four cases can occur; they do not indicate how the framework could be applied to improve the robustness of a process.

Table 1. Chalupnik et al. (2008) identify arrangements of tasks which act to propagate, accumulate, absorb or reduce task delays.

Pattern	Process delay / task delay	Likely effect on process robustness
Propagation	*Between 0 and 1*	Small decrease
Accumulation	*>1*	Large decrease
Absorption	*0*	Small increase
Reduction	*<0*	Large increase

In the second category, other authors use numerical/analytical methods to study design process robustness by focusing on particular case studies. For instance, Yassine (2007) simulates a model of an automotive hood development process and shows how this process could be re-engineered to improve its robustness, which he describes as its ability to absorb unexpected design change. Chalupnik et al. (2008) describe the construction of a 225-task model of an aero-engine design process and the subsequent application of simulation methods to examine the process' robustness (Figure 1; simulation results modified to protect commercial sensitivity). They conclude that models of design processes on this scale typically comprise too many uncertain variables to easily investigate their robustness. Furthermore, the application of simulation modelling to analyse a system's response to uncertainties in its input variables presupposes confidence in the accuracy of the model. In the case of complex design processes, however, it is difficult to calibrate simulation models or otherwise compare them to the process they represent. This difficulty arises from factors including the long duration and non-repeatability of projects, the complexity of process behaviour and the limited overview of process participants. The need to minimise the cost of modelling also necessitates many simplifying assumptions, for instance in how a project responds to the discovery of rework (Wynn et al., 2007).

In summary, therefore, much design process robustness literature either provides high-level descriptions of influencing factors or is based on the analysis of high-fidelity process models which can be difficult to obtain in practice. This paper complements the existing literature by showing how simulation methods can be applied to explore the determinants of design process robustness without requiring simulation models of specific processes. We argue that these insights can be applied to improve more complex, realistic processes.

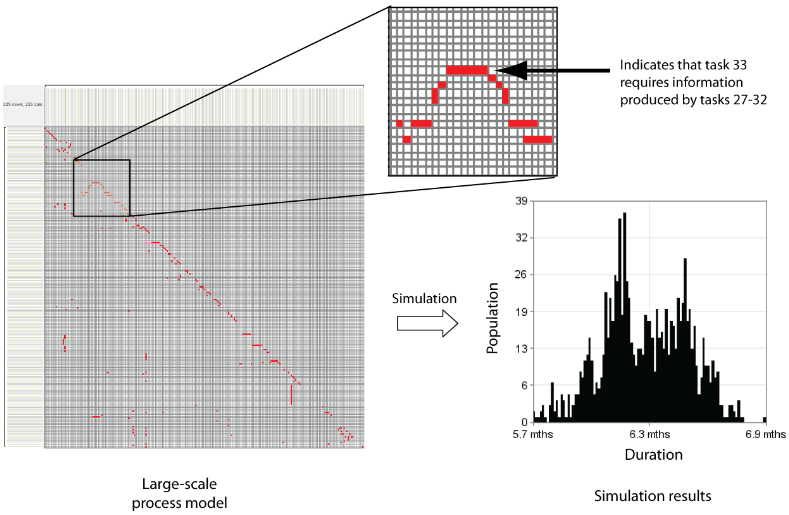


Figure 1. Analysing the robustness of specific processes to uncertain input variables requires a high level of confidence in model fidelity and is computationally expensive

In this paper, process simulation is used to investigate how the robustness of task clusters is influenced by the number of dependencies between tasks and the resource constraints within the cluster. We show how heuristic conclusions can be drawn about the factors which govern process robustness, and argue that these heuristic results can provide insight to support the improvement of more complex processes. In particular, statements such as 'modifying a process to reduce the interdependencies between tasks tends to improve its robustness' can be developed from simulation results and can provide guidance for improvement which does not require a high-fidelity model of a specific process or computationally expensive analysis of its behaviour. Furthermore, since such heuristics do not require knowledge of the specific organisation of tasks in the process, they could apply to the design of novel products for which the process cannot be easily described and is therefore difficult to simulate directly.

Our approach is similar to Flanagan et al. (2007) who study how the connectivity of tasks in a process can influence its likely range of durations. They simulate six hypothetical processes, each comprising 100 tasks whose concurrency is limited by a different number of randomly-placed information dependencies. They conclude that as this number increases from 5% to 30% of the theoretical maximum (in which every task requires information from every other task) the variability of possible durations also increases. They describe the processes with increased variability as less robust than the baseline case. These findings offer a possible route to improve process robustness, i.e., by re-engineering design processes to reduce the information flow constraints. However, the generality of Flanagan et al.'s conclusion is limited by the small sample of models which were investigated.

Instead of simulating a small number of hypothetical models of realistic complexity, in this paper we attempt to address the sample size issue by simulating large numbers of hypothetical fragments comprising 3, 4, 5 and 6 tasks respectively. Since design processes are in practice partitioned into groups of tasks which interact through relatively well-defined interfaces, we propose that it is possible to draw useful insights by examining the statistics of how large numbers of hypothetical fragments behave. We model the clusters using a task-based simulation framework, in which the sequencing and concurrency of tasks is constrained by the information flows and resource constraints between those tasks. We generate all possible dependency networks for 3, 4, 5 and 6-task clusters then assign a range of values for the duration of each task in each such variant. Each variant is then simulated to calculate its response to a delay introduced to each task in turn, relative to the baseline case of no delays in that variant. The results of these experiments are presented in a form which highlights the impact of task connectivity degree and resource constraints upon the ability of the cluster to absorb delays, independent of other variables which influence process behaviour. The experiment indicates that the findings are largely independent of the number of tasks for the 3, 4, 5 and 6-task cases.

2. Assumptions

The experiments are based on a process modelling framework which allows process duration to be calculated from the properties of individual tasks and the structure of information flows between tasks. This framework is called the Applied Signposting Model (ASM) and is described in full by Wynn et al. (2006). In overview, the ASM is based on the following assumptions:

- The order of attempting tasks is governed by information flows, such that a task is attempted immediately that all its predecessors are completed.
- Tasks may be possible to execute concurrently given the information flow constraints. However, this may be limited by resource availability – if two tasks are ready to start but both compete for the same resource, then one must be selected and attempted first. When that task is completed, the resource is released and the second may be attempted.
- 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. This is illustrated in Figure 2, which shows an example scenario in which a policy of selecting the longest task first whenever conflicts arise (Figure 2a) leads to a longer total duration than a policy of selecting the shortest task first (Figure 2b).
- Tasks are independent during execution. In other words, a task's total duration and resource utilisation are determined at the time it is started and cannot be influenced by subsequent events.
- Tasks may have several possible output scenarios. When the task is completed, a single output is selected according to logic specified in the task definition (for instance, an evaluation task may 'fail' with a specified frequency). This allows branching and iteration. A number of other assumptions govern the behaviour of the model in response to rework. However, since the experiments reported in this paper do not examine iteration these assumptions are not discussed here.

The ASM incorporates Monte-Carlo simulation code to determine the distribution of durations for a given process model from these assumptions. Simulation is a widely-accepted method for exploring the behaviour of a system based on a formal model of that system. Monte-Carlo methods are particularly suited to analysis of models containing a large number of variables or which cannot be easily evaluated through formal analytical methods. The latter is the case in this paper, where we apply simulation to estimate the duration of resource-constrained concurrent processes.

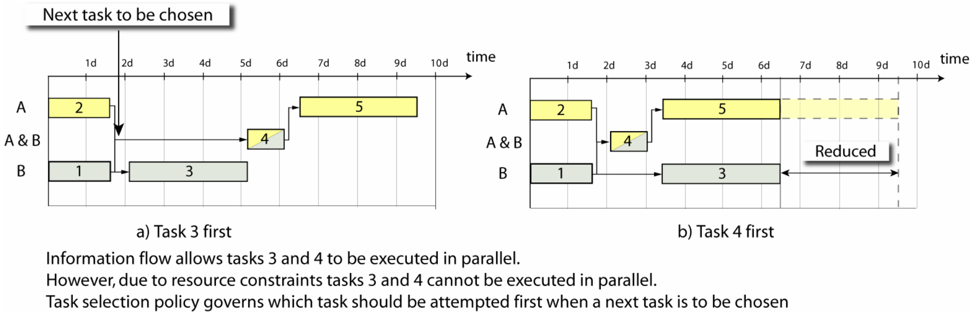


Figure 2. An example of two task selection policies applied to a process with resource constraints (requirement of each task for one or both of resources A and B is indicated on the vertical axis)

3. Objectives

The experiments reported in the remainder of this paper have two objectives. Firstly, we examine the factors which impact upon robustness of engineering design processes – more specifically, we explore what determines how delays in completing individual tasks lead to greater or lesser delays in the overall process. Secondly, and more generally, we explore how the simulation of hypothetical process clusters can provide insights to support process improvement.

Given the modelling assumptions outlined in Section 2, the effect of a delay within any cluster of tasks upon the cluster's total duration is determined by the following variables:

- the number of tasks in the cluster;
- the duration of each task;
- the information flows which constrain the order in which tasks can be attempted;
- the resource constraints which can prevent tasks from being executed concurrently;
- the task selection policy;
- the task(s) whose completions are delayed;
- the duration(s) of the delays.

Although the hypothetical fragments approach could be applied to examine the influence of any of these factors on process robustness, in this paper we concentrate on the following question:

- What are the impacts of information flow density and resource constraints upon the robustness of a cluster's duration to delays in constituent tasks?

We treat all other factors as uncertain variables which fall within given ranges. This allows the impact of information flow density and resource constraints on process robustness to be studied independently of other factors in the model.

4. Method

Two simulation experiments were conducted to explore the question outlined above, focusing respectively on processes where concurrency is governed by information constraints alone and those which are also influenced by resource constraints:

4.1 Processes without resource constraints

The following steps were executed for each of N = 3, 4, 5 and 6 tasks:

1. Generate clusters by assigning tasks' durations

The durations of tasks 1-N in the cluster are each assigned a value selected at random from the set [1, 2, 3, 4] in days.

2. Generate all non-redundant dependency constraints on concurrency

It is assumed the tasks in the cluster are executed in the order they are generated and with no iterations; however, the information dependencies between the N tasks constrain which can be performed concurrently. The set of all possible constraints on concurrency is generated using simple combinatorics (Figure 3).

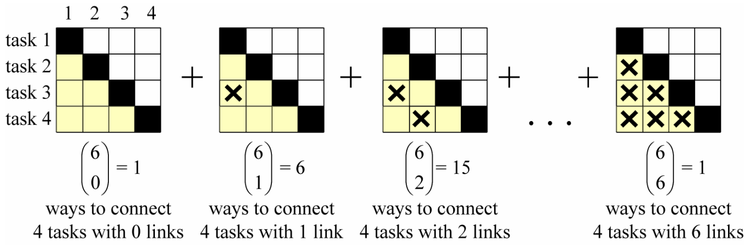
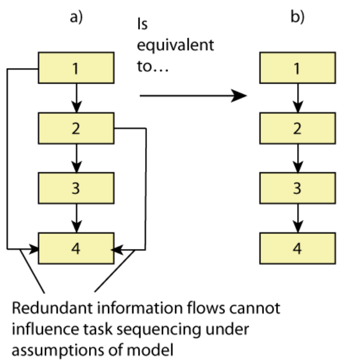


Figure 3. Computing possible information flow constraints on concurrency in a 4-task cluster

The total number of information flow dependencies n_{Total} can thus be calculated from the number of tasks $nTasks$ using the following formula (the `number of graphs on n labelled nodes):`

$$n_{Total} = 2^{nFields}, \text{ where } nFields = \frac{nTasks*(nTasks-1)}{2} \tag{1}$$

This shows that the number of variants increases exponentially with the number of tasks in a cluster. However, many of these variants may exhibit identical behaviour, since they contain redundant links which may be removed without affecting the possibilities for concurrency. In terms of information flows, these 'redundant links' arise from the feed-forward of information through the process. This is shown by example in Figure 4 (left).



No. of tasks	Total number of process variants (n_{Total})	Number of variants with no constraint redundancy (n_{Unique})	$\frac{(n_{Unique})}{(n_{Total})}$
3	8	7	0.875
4	64	40	0.625
5	1024	357	0.349
6	32768	4824	0.147

Figure 4. A transitive reduction algorithmis used to identify processes with redundant constraints and remove them prior to simulation

Those variants with redundant concurrency constraints were removed prior to simulation. This was achieved using a standard graph-theoretic algorithm for transitive reduction (Aho et al. 1972, implementation by Cotton 2007). As a result, in the next steps of the experiment only non-redundant dependencies were considered. This significantly reduced the computational expense of the experiment (Figure 4, right) and thereby allowed the analysis of 6-task clusters which would otherwise have not been possible.

3. Introduce task delays of 50% on each task in turn and simulate variants

After assigning durations and concurrency constraints, the model was simulated using the Monte-Carlo algorithm of the Applied Signposting Model. As all task durations are deterministic values selected from the set [1,2,3,4] days and there are no resource constraints, there is no uncertainty in process outcome and only one simulation run is therefore required to calculate the precise outcome.

Each variant of the cluster was simulated *number of tasks+1* times, once for the baseline configuration with no delays and once for each of the configurations with one task delayed by an additional 50% of its original duration.

4. Repeat Steps 1-3 to account for the effect of different task durations

To account for the possible effect of different task durations upon process response to delays, Steps 1-3 were repeated (on each repetition, a different set of durations was thus selected in Step 1). For the 3, 4 and 5 task clusters the experiment was repeated 10 times; for the 6-task clusters it was repeated 3 times to limit computational expense.

5. Repeat Steps 1-4 to account for the effect of different delay durations

To assess the influence of the duration of a task delay on a cluster's response, Steps 1-4 were repeated but delays of only 25% (instead of 50%) were introduced in Step 3.

4.2 Processes with resource constraints

The experiment was then modified to explore the impact of resource constraints upon cluster behaviour. It was assumed that 2 different resources A and B were available. For instance, if a task was specified as requiring Resource A it could not be attempted in parallel with any other task requiring Resource A – even if the information dependencies would allow concurrency.

For each of N = 4 and 5 tasks, a cluster with N tasks was generated and the following steps executed (Steps 1, 2 and 4 are equivalent to those described above):

1. Generate clusters by assigning tasks' durations

2. Generate all non-redundant dependency constraints on concurrency

2a. Generate all possible resource constraints on each variant

Each information flow-constrained cluster variant is then supplemented with all possible combinations of resource requirements for tasks in that cluster. This assumes that each task required either one or both of the resources A and B in order to execute. The total number of graphs considered in Step 3 was thus significantly increased from the first experiment; to limit computational expense the experiment was thus only considered for 4- and 5-task clusters, and only delays of 50% were considered.

3. Introduce task delays of 50% on each task in turn and simulate variants

When two or more tasks are possible to attempt concurrently but this is not possible due to resource constraints, one task is chosen at random and selected to begin first. When that task is completed, the resource becomes available again and the next task may be attempted. This random task selection reflects uncertainty in the way these decisions are made in practice and introduces the possibility of variability in the response of any simulated cluster variant. Therefore, more than one simulation run is required to assess each variant's behaviour.

50 simulation runs were therefore performed for each simulated variant. If more than one possible duration had emerged by this point, an additional 500 simulations were run and the total duration was taken as the mean of all durations revealed through simulation – otherwise, it was assumed that the variant's duration was deterministic.

4. Repeat Steps 1-3 to account for the effect of different task durations.

5. Results

To facilitate presentation of the results, the discrete categories of process robustness outlined by Chalupnik et al. (2008) and summarised in Table 1 were generalised into a continuous metric termed the *robustness angle* ρ *(Equation 2)*. This is a measure of the amount by which a delay of some specific magnitude is absorbed (in a more robust process) or propagated (in a less robust process):

$$\rho = \operatorname{atan}\left(\frac{t_i - b}{c_i}\right) \qquad (2)$$

where:
t_i - cluster's duration after i-task is delayed;
b - cluster's baseline duration;
c_i - i-task's delay.
This metric allows comparison of processes according to their degree of robustness and thereby enables more detailed analysis than the discrete categorisation framework.

5.1 Robustness plot and robustness profile

The metric introduced above allows the robustness of a cluster to be summarised graphically using a *robustness plot*. For instance, consider a hypothetical cluster comprising 5 tasks. A delay of 0.5 days to Task #1 might lead to a process delay of 0.5 days, thus placing the point *P5* on the 45° line of the plot in Figure 5. Within the same cluster, a delay of 0.5 days to Task #2 could be absorbed, leading to the point *Ab* on the 0° line. For each such task delay which is considered in the analysis, another point will be added to the plot. For this 5-task cluster with a given connectivity and a delay of 0.5 days in any one of the five tasks, five points would therefore appear on the map as shown in Figure 5.
This example highlights that a given cluster does not have a single value of the robustness angle, but has a range of responses according to the factors considered in the analysis. In this case, the plot should be interpreted as the response of the specific 5-task cluster which is precisely specified apart from the origin of the delay. The way process duration is affected by delay is contingent upon a number of other factors as well as the delay's origin. For instance, shorter delays might be less likely to propagate/accumulate than longer delays where a task is not originally on the critical path. If included in the example outlined above, this would appear as 5 points each to the left and below of those shown on the plot. Conversely, a longer delay to each task might cause propagation even if the task was not originally on the critical path. In such cases, the robustness angle would be expected to increase with delay duration, all other factors remaining equal. To illustrate, the hypothetical change in point P2 as the delay is varied is illustrated by the dotted line in Figure 5.

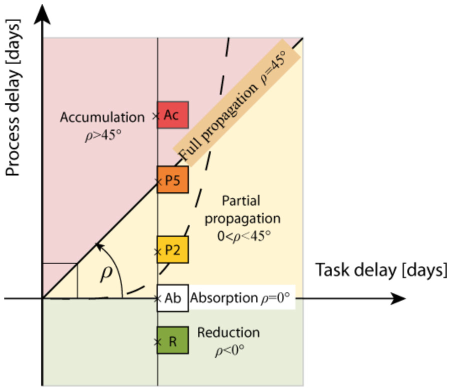


Figure 5. The robustness plot is a graphical representation of the relationship between the magnitude of task delay (cause) and the magnitude of process delay (effect)

The robustness plot is a useful summary of a process' response to different delay magnitudes. As more points are calculated and plotted for different assumptions regarding the cluster's configuration and the delay, it may be viewed as a density plot. If the magnitude of delay is not of specific interest, the density plot may be integrated radially and presented as a *robustness profile* (Figure 6). This alternative representation indicates the relative frequencies of different ranges of the robustness angle and thereby highlights the robustness of the cluster with respect to all factors which were considered in the analysis which generated it. For instance, the profile in Figure 6 indicates that the cluster is either in the 43° to 49° range (full propagation, labelled *P5*) or in the -6° to 6° range (absorption, labelled *Ab*) with a small possibility of partial propagation (labelled *P2*). On the robustness profile, a 'more robust' cluster is then more heavily weighted towards the left hand side of this profile.

In the following two subsections, the results of the experiment are presented as robustness profiles prior to highlighting the insights which were drawn from the simulation method.

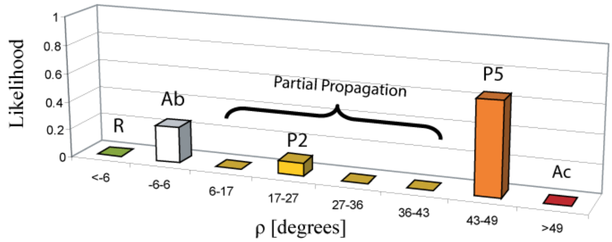


Figure 6. The robustness profile summarises the robustness of different processes in terms of the relative frequency of each angle, without distinguishing between factors which impact upon this

5.2 Robustness profiles for clusters without resource constraints

Figure 7 shows the results of the simulation experiment outlined in Section 4.1. This shows that in the distributions of the robustness angle for clusters with a 25% task delay, only two values of the angle appear: 0° and 45°. In other words, in the clusters generated during the experiment, a 25% delay to any non-critical task was insufficient to influence the critical path.

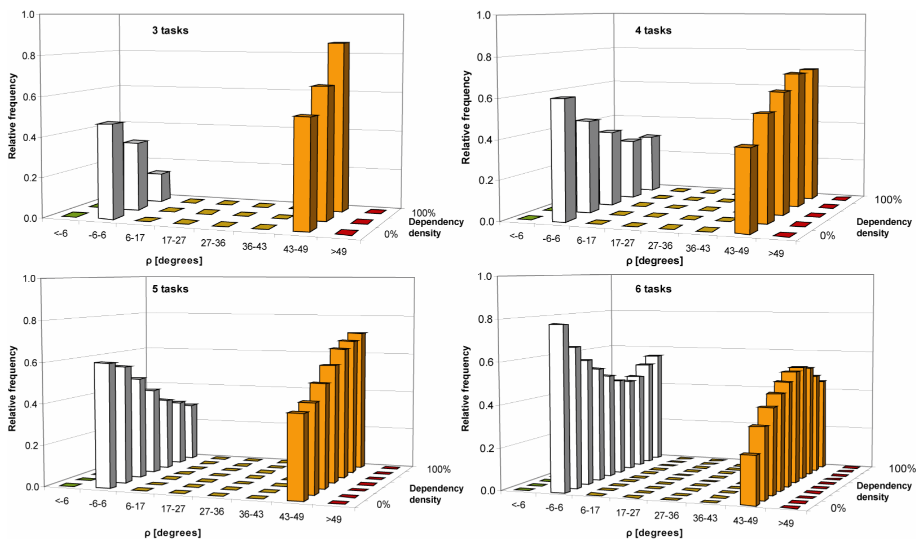


Figure 7. Simulation results showing the effect of a 25% delay in one task in a 3,4,5 and 6-task cluster with varying dependency density

Increasing tasks' delays to 50% of their base durations leads to an additional bar appearing in the profiles (Figure 8). The new bar is centred around 26.5° and represents partial propagation of the delay – however, in all the 3, 4, 5 and 6-task experiments this is significantly less likely to occur than the full propagation or absorption cases. Regardless of the magnitude of delay, Figures 7 and 8 show that the relative frequencies of absorption, partial propagation or full propagation cases are influenced by the degree of task connectivity (0% – no dependencies between tasks, 100% – maximum number of unique task dependencies). A trend is visible in all the graphs: for low values of connectivity, the higher the degree of connectivity, the more likely the cluster is to propagate the task delay regardless of the number of tasks in the cluster and of the magnitude of delay. In other words, as non-redundant edges are removed, a cluster could be viewed as becoming more robust, in that it becomes more likely to have a lower mean value of the robustness angle.

5.3 Robustness profiles for clusters with resource constraints

Figure 9 shows the results from simulating the resource-limited processes, and highlights a trend of decreasing robustness with increasing connectivity which is similar to the non-resource-limited cases above. However, by adding resource constraints to the models, the robustness for any given connectivity is reduced – as indicated by the higher bars of the 43° to 49° range (rightmost but one row) in Figure 9 compared to the equivalent bars in Figure 8. In addition to the three values observed in the non-resource-limited cases, several other values of the robustness angles may be seen. However, the corresponding cases of accumulation and partial propagation remain significantly less likely that the dominating cases of absorption, 26.5° propagation and full propagation.

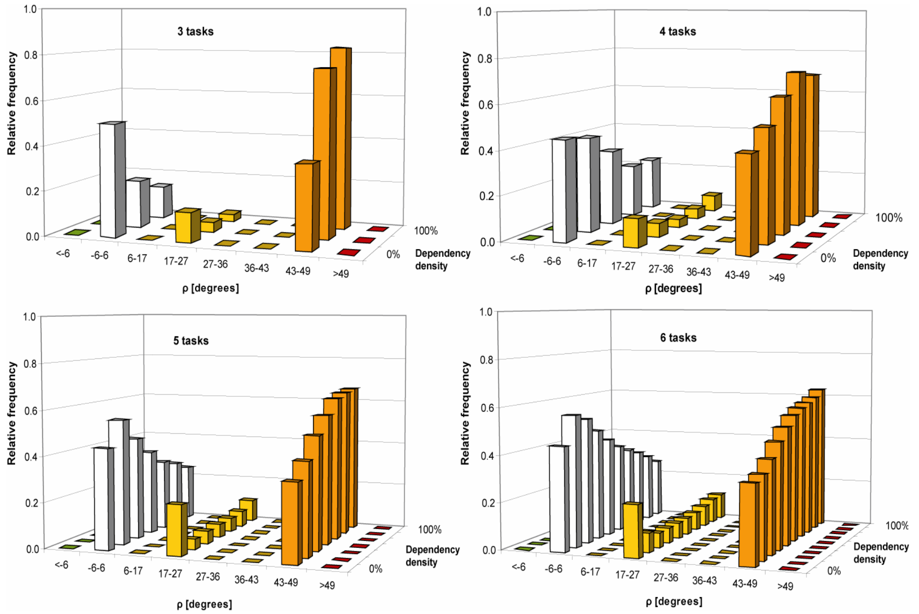


Figure 8. Simulation results showing the effect of a 50% delay in one task in a 3, 4, 5 and 6-task cluster with varying dependency density

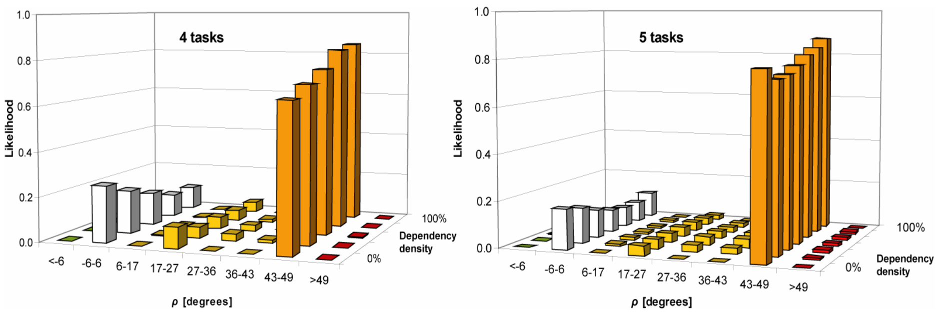


Figure 9. Simulation results showing the effect of a 50% delay in one task in a 4 and 5-task resource-constrained cluster with varying dependency density

6. Discussion

This section discusses the results of the simulation experiment under two headings. Firstly, insights drawn from the specific experiments are discussed. Secondly, we reflect upon the hypothetical fragments approach and suggest opportunities for further research to refine this method.

6.1 Discussion and limitations of the robustness experiment

The following general insights and heuristics were drawn from the robustness experiment:

1. **Effect of connectivity density on robustness.** The results show that removing non-redundant information dependencies increases the robustness of a 3, 4, 5 or 6-task cluster in the average case. Given the experiment design, this result would be expected because removing a non-redundant constraint always allows more tasks to be executed in parallel and thus fewer of them are on the critical path. However, the experiment has quantified this belief by indicating the degree to which certain types of cluster would become more robust following a reduction in density. The results also indicate that the impact of removing a non-redundant link is dependent upon the link density in the process; in terms of Figures 7 and 8, the slope of the 43° to 49° bars reading into the page is not constant. Unexpected behaviour was revealed in the 25% delayed 6-task case (Figure 7, bottom-right plot) – in this case, the robustness of the process may be seen to increase with dependency density for high density values.
2. **Effect of delay magnitude.** Increasing the duration of a task delay from 25% to 50% of the baseline task duration was shown to result in some of the delays being partially propagated. This may be explained by the fact that following a delay, some tasks that were originally not on the critical path can become critical such that any additional delay would lead to propagation. As above, the simulation experiment has quantified this heuristic by showing that the effect is relatively infrequent in the 3, 4, 5 and 6-task clusters – the relative frequency of absorption and full propagation are much more important in terms of determining the process behaviour.
3. **Effect of resource constraints on robustness.** The results show that adding resource constraints has a similar (i.e., negative) effect on the robustness of a cluster as increasing the task dependency density. This is expected since a resource conflict between two or more tasks prevents them from executing concurrently, in a similar fashion to information dependencies between those tasks.

In contrast to the non-resource constrained experiments, the responses of each cluster include cases spread across the entire range of positive robustness angles – including a small number of accumulation cases (small red bars in Figure 9, right). The results highlight these effects and show that they do not have significant impact on the robustness of those clusters we analysed. Further work is needed to fully explain the mechanisms by which this occurs.

1. **Relative frequency of reduction and accumulation patterns.** The results indicate that reduction and accumulation patterns occur with only negligible frequency. However, we believe this could be an artefact of the simulation experiment and further work is required to explore this finding. In particular, for the resource-constrained cases each variant was assigned a robustness angle calculated from the mean value of its simulated duration. Thus, if simulation of a particular cluster under a specified delay to a particular task revealed that in some of the possible task orderings (which are all sampled during simulation due to the random task selection policy) it behaved as a reduction pattern and in others it exhibited full propagation, the cluster would be categorised in the partial propagation range. Due to this effect, any cluster variants which could exhibit reduction under some circumstances might appear in the partial propagation range of the robustness profiles.
2. **Robustness metrics.** The results highlight that studying the distribution of a cluster's response to task delays using the robustness profile can lead to insights which could easily be overlooked if robustness was described using simpler summary metrics. For instance, the profiles shown above indicate that cluster variants with similar likelihoods of absorbing a task

delay can have significantly different distributions over the remaining range of robustness angles.

In summary, therefore, the experiments have shown that the results of the hypothetical fragments simulation are consistent with heuristic expectations derived from critical path theory. They have also shown how the hypothetical fragments approach can quantify these heuristics by indicating how often and under what circumstances they can be expected to hold.

The simulations reported in this paper illustrate the hypothetical fragments approach and explore the impact of resource constraints and information flows on process robustness. However, a number of experimental limitations would need to be addressed in order to justify the extraction of specific guidance for robustness improvement. Firstly, more careful experiment design would be required to justify the choice and number of experiments included in the analysis. Secondly, the effect of introducing delays into multiple tasks simultaneously should be considered. Thirdly, a key characteristic of design processes in practice is iteration. This was not included in the experiments due to the increased computational expense of simulation it would incur, as well as the need to consider which of the many models of rework behaviour found in the literature (see Wynn et al., 2007) would be most appropriate to incorporate.

6.2 Discussion and limitations of the hypothetical fragments approach

To recap, the hypothetical fragments approach uses simulation to calculate how a model of process behaviour responds to many different combinations of the model's input variables listed in Section 3. These experiments are executed for many small clusters of tasks; it is assumed that the behaviour of these clusters can provide insight into more complex, realistic processes for which accurate simulation models are difficult to obtain. The insights are developed by aggregating simulation results to indicate the likelihood of certain effects occurring independently of the input variables, apart from certain metrics of specific interest – in this paper, we examined the dependency of design process robustness upon task connectivity and resource constraints.

One limitation of this approach is that the aggregation of results makes assumptions about how probable each value of each input variable is to occur in practice. For instance, it was assumed in the robustness experiments that a delay is equally likely to occur in any task in a given variant. However, in practice this *a priori* distribution of causes might not be uniform – for instance, it could be argued that tasks which require input information from many sources are more complex than those with fewer predecessors, and thus more susceptible to delays in completion. In other words, the hypothetical fragments approach allows process behaviour to be examined under certain *a priori* assumptions – but these assumptions must be carefully considered when interpreting the results. Two opportunities for further work arise to support this consideration. Firstly, empirical work could be conducted to investigate the form of the *a priori* distributions in practice – for instance, what factors impact upon the likelihood of a task being delayed and how are these factors related to the task's context in the process? Secondly, additional simulations could be conducted to investigate the sensitivity of a given experiment to the chosen *a priori* distributions and the experiment design.

Another limitation of the hypothetical fragments approach is its computational expense which prevents application to large clusters of tasks – hence it is necessary to assume that insights drawn from small clusters are applicable to more complex, realistic scenarios. We propose two ways in which this assumption could be explored. Firstly, more complex processes can be viewed as a composition of smaller 'building blocks'; further research could explore the degree to which findings regarding small clusters of tasks can be generalised to provide insight into processes of greater complexity. This would draw upon knowledge of how such processes are structured in practice; for instance, complex processes are typically partitioned by stage-gate systems into sequenced sub-processes to support project management and control. Secondly, the exhaustive enumeration of possible information flow constraints used in this paper could be replaced with a Monte-Carlo sampling approach (random or guided sampling from the set of all possible dependencies between the tasks in a larger model). This would allow more complex models to be investigated directly.

7. Conclusions

This paper has shown how simulation experiments can be used to investigate how the robustness of small clusters of tasks depends on their composition, and argued that this can provide insights to support the improvement of more complex processes.

In summary, two main contributions were made. Firstly, an approach was proposed to explore the determinants of process behaviour through simulation of hypothetical fragments. This allows heuristics about process behaviour to be quantified by calculating how frequently and under what circumstances they could be expected to hold, without requiring a detailed model of a specific design process. However, the results of such analyses must be interpreted with care as they depend on the scope and weighting of possible processes considered by the simulation experiment.

Secondly, the approach was applied to explore the dependency of process robustness upon the connectivity and resource constraints of the process. The results suggested that reorganising a process to reduce the number of dependencies can decrease the likelihood of a task delay propagating to delay the entire process, and that reducing the number of resource constraints (e.g. by cross-training personnel) can have the same effect. The experiment showed this only for small clusters of 3, 4, 5 and 6 tasks; however, since more complex processes may be viewed as composed from arrangements of such clusters, future work will aim to explore how these findings could be generalised.

References

Aho, A.V., Garey, M.R. and Ullman, J.D. (1972) The transitive reduction of a directed graph. SIAM Journal on Computing 1(2) pp. 131-137.

Chalupnik, M.J., Wynn, D.C., Eckert, C.M. and Clarkson, P.J. (2008) Investigating design process performance under uncertainty, 8th International Symposium on Tools and Methods of Competitive Engineering (TMCE 2008), Izmir, Turkey.

Cotton (2007) http://www.verimag.imag.fr/~cotton/

Flanagan, T.L., Eckert, C.M. and Clarkson, P.J. (2007) Process structure: an important factor in design project planning? 16th International Conference on Engineering Design, ICED'07, Paris, France.

McManus, H.L. and Hastings, D.E. (2005) A framework for understanding uncertainty and its mitigation and exploitation in complex systems. in 15th Annual International Symposium of the INCOSE.

Whitney, D., Crawley, E., de Weck, O., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J. and Wallace, D. (2004) The Influence of Architecture in Engineering Systems, in 1st MIT Engineering Systems Symposium, MIT, Cambridge, MA.

Wynn, D.C., Eckert, C.M. and Clarkson, P.J. (2006) Applied Signposting: A modeling framework to support design process improvement. Proceedings of IDETC/CIE 2006 ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, September 10-13, 2006, Philadelphia, Pennsylvania, USA. DETC2006-99402.

Wynn, D.C., Eckert, C.M. and Clarkson, P.J. (2007) Modelling iteration in engineering design. 16th International Conference on Engineering Design, ICED'07, Paris, France.

Yassine, A.A. (2007) Investigating product development process reliability and robustness using simulation, Journal of Engineering Design 18(6) pp. 545-561.

Marek J. Chalupnik,
Engineering Department,
Trumpington Street,
Cambridge. CB2 1PZ. United Kingdom