MODELLING THE INFLUENCE OF UNCERTAINTY, PROCESS ARCHITECTURE AND FEEDBACK DYNAMICS ON PD PROJECTS

H. Nam Le, David C. Wynn and P. John Clarkson
Engineering Design Centre, University of Cambridge

Keywords: design structure matrix, system dynamics, model transformation, uncertainty, process robustness

1 INTRODUCTION

Managing product development (PD) processes is complex due to the many factors that influence process behaviour and performance. These factors can stem from different process abstraction levels and may be related to process architecture or high-level project issues. One way to analyse and optimise a process considering all these viewpoints is to represent the variety of influences from different process abstraction levels within a single modelling framework. In this paper, an method to achieve this by combining Design Structure Matrix models with System Dynamics is presented. The potential of the method is demonstrated through analysis of a hypothetical PD process. This analysis shows how it is possible to study the combined influence of process architecture, as represented in a DSM, and high-level project and management strategies, as represented in the feedback loops of an SD model. Furthermore, a typical source of uncertainty found in many PD projects – uncertainty regarding the duration of individual activities – is incorporated in the combined model to show how it is possible to improve the robustness of recommendations derived from simulation results. It is shown how managers may use this method to narrow down the possible range of their action.

2 APPROACH

Process models are an abstract representation of real processes, constructed from a certain perspective for a specific purpose. Depending on this purpose, process models may adopt different views and abstraction levels. This paper focuses on the combination of two commonly applied modelling approaches, Dependency Structure Matrix (DSM) and System Dynamics (SD), which take different abstraction levels. We show how combining these two approaches allows consideration of multiple perspectives and issues concurrently.

DSM models of PD processes represent activities and their interdependencies in a concise matrix format, allowing the modeller to describe the architecture and details of a PD process on the task level (Browning et al., 2005). In terms of architecture, the model can represent the process, its sub-processes and constituent tasks. By viewing tasks as information processing units, DSM models show information flows between activities. These dependency relationships have a significant influence on the course of the process because they determine the tasks which are possible to attempt at any state of project progress. In terms of process details, DSM models can include information about task durations and resource requirements, and can represent uncertainty in task duration as well as the likelihood of iterations occurring (Carracosa et al., 1998). These process details and architecture-related characteristics, as captured in a DSM model, are useful in identifying iterations in the PD process and exploring their potential impact on process lead time.

SD models, on the other hand, view processes from a higher abstraction level than DSM models. In SD, PD processes are modelled as collections of identical work packages (WPs). WPs are assumed to flow at certain 'rates' between 'stocks' that indicate the current execution state of each WP, according to a generic rework cycle which can be complemented by feedback loops that govern how flow rates change over time (Cooper et al., 2002). One of the main factors that determine the flow rates in SD rework cycle models, such as that described by Ford and Sterman (1998) is the so-called 'process concurrence relationship'. This specifies the percentage of WPs which are possible to attempt at every possible state of project progress (or it can alternatively be formulated as the percentage of WPs already complete at each time in the project). The standard SD rework cycle is used to model undiscovered rework due to WPs that are believed to be done but in fact contain undiscovered errors, and to investigate how this false belief may influence process behaviour when these errors are discovered at a later stage. Apart from rework discovery, influences imposed on process behaviour by project constraints such as resource size and project deadlines and management policies are modelled as elements of dynamic feedback systems. These feedback systems allow the study of causal relationships between influencing factors and process behaviour, and the design of policies to change this behaviour for better process performance (Lyneis and Ford, 2007). Thus, SD models are useful to explore and understand the dynamics of process behaviour on an overall-system level, providing a means to investigate issues that cannot be easily represented in a DSM model. However, in some respects they are quite abstract since they do not directly account for the structure of dependencies and information flows in a particular process.

This paper shows how the two modelling approaches described above can be integrated in order to capture both the high-level project issues described by SD models and the task properties/task dependency information represented in DSM models. This provides a simulation and analysis framework to explore process behaviour in a more holistic way than either approach can provide in isolation. In an earlier paper (Le et al., 2010) we present a method to integrate these two modelling approaches by transforming a flow-chart style task network model into the rework cycle of an SD model. This paper extends the initial result by (1) applying the method to a DSM process model and (2) showing how uncertainty in task duration can be incorporated in the analysis. In particular, different task durations can influence the order in which activities are executed, resulting in a set of different project schedules and lead times. Thus, accounting for uncertainty in task duration and its associated influences can improve the robustness of insights derived from process simulation.

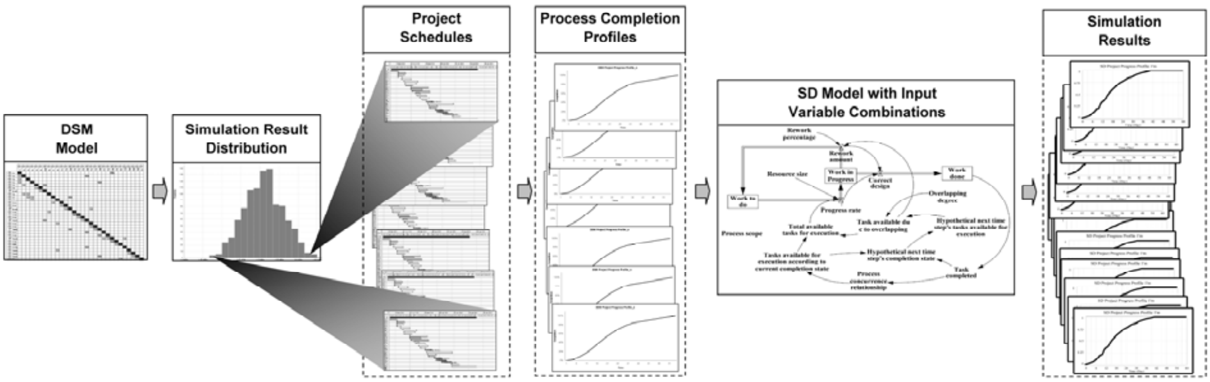


Figure 1. Integrated modelling and simulation framework

Figure 1 illustrates our integrated modelling method on a high level. A DSM model is used as the starting point to capture task-level details of a specific PD process model from the information dependency perspective. The second step is to automatically transform the DSM process architecture into rework cycles of an SD model. In particular, the sub-processes of the DSM model are transformed into individual rework cycles in the SD model. Dependency relationships between tasks in the DSM sub-processes are used to determine the process concurrence relationships in each SD rework cycle. The derivation of process concurrence relationships can be achieved as follows. First, the DSM model is simulated to obtain a lead time with a specific task schedule that can be visualised as a Gantt chart (no resource constraint is assumed to avoid schedule distortion). This chart represents a variant of possible task orders evolving from the model underlying dependency relationships. Second, by knowing the start and end dates of each task depicted in a chart, it is possible to calculate (1) the individual and (2) the cumulative percentage contribution of each successive time step to the overall project scope achieved in terms of number of tasks completed. Then, by equating a time unit of DSM task duration with a WP in SD terms, process concurrence relationships of an SD model can be derived from the process completion profile (PCP). These relationships specify the percentage of WP available for attempt according to a given progress state at each time step (for more details of the transformation method see (Le et al., 2010). Finally, once a rework cycle (with its process concurrence relationships) is generated, the SD model can be completed by adding SD model elements to the rework cycle. These include management policies and other feedback systems (Ford and Sterman, 1998).

Apart from process architecture-related characteristics, the model transformation method considers details of activities captured in the DSM which influence the course of the PCP. For instance, variabilities in task duration yield different task execution orders and process lead times, which are revealed when multiple simulation runs are executed. In order to account for such variability, which can be modelled through probability density functions associated with each task, Monte-Carlo simulation of the process as represented in the DSM is used to obtain a probabilistic distribution of different project lead times with different task schedules that could be visualised as Gantt charts. Each outcome provides a different process completion profile. By converting every PCP resulting from DSM simulation into a specific process concurrence relationship representing that outcome, then generating and simulating the SD model for each concurrence relationship, it is possible to explore (1) how the process could behave under task duration uncertainty, and (2) the set of SD input variable combinations that are more robust than others, i.e., able to deliver satisfactory process performance In the face of this uncertainty.

The method outlined above is currently implemented using a combination of automatic and manual analysis. A fully-automated implementation is currently under development as a plug-in for the Cambridge Advanced Modeller platform (formerly P3). Once complete, this should allow fast and seamless application of the transformation to any process modelled as an activity DSM in that software.

3 CASE STUDY

To illustrate how the method can be applied to investigate the impact of multiple influencing factors from different process abstraction levels, a hypothetical process model with twenty four tasks was synthesised (Figure 2, left). In this hypothetical model, each task was assigned a triangular probability distribution function representing its duration. Project schedules generated from the DSM model through Monte-Carlo simulation were then used to calculate project completion profiles, from which process concurrence relationships of rework cycles in the SD model were derived as outlined in Section 2.

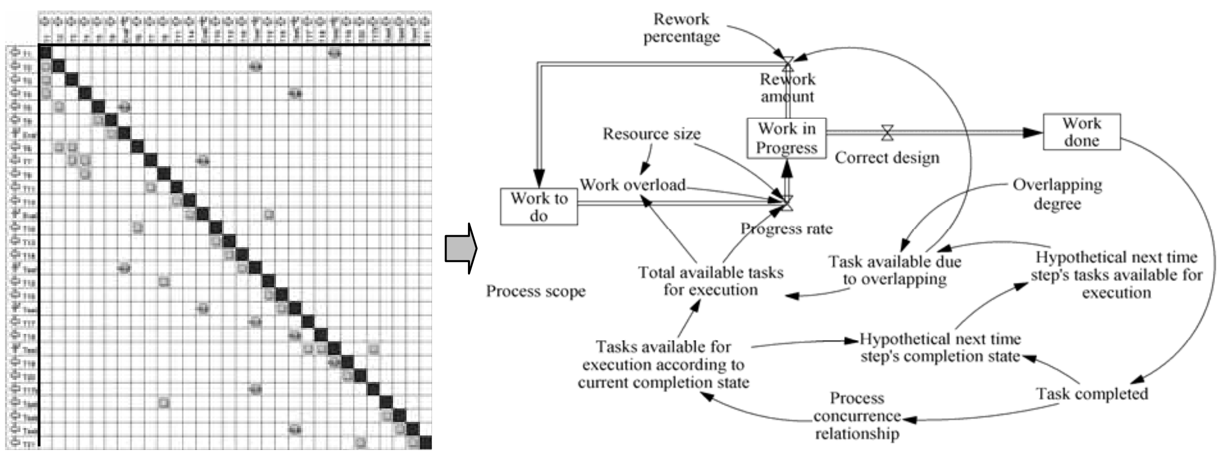


Figure 2. Models of the hypothetical process

In the resulting SD model shown in Figure 2 (right) the variables 'Resource size' and 'Overlapping degree' are modelled as management levers (input variables), with the resource size varying between 4 and 7 and the overlapping degree ranging from 15% to 50%. It is assumed that overlapping of tasks results in rework whose amount is characterised by an exponential function, implying that the later the overlapping starts, the lower the proportional amount of work package that needs rework, and *vice versa*. Furthermore, it is assumed that work pressure arises when the number of tasks available for execution is greater than the capacity provided through the resource size. This has the negative impact of a decrease in resource efficiency, resulting in longer work package processing time (Figure 2, right). Based on these simple assumptions the SD model simulates all possible combinations of the two input variables – resource size and overlapping degree – for all the process concurrence relationships calculated from each project schedule derived from Monte-Carlo DSM simulation.

Figure 3 (left) presents the results from all simulation runs of this SD model in a parallel coordinates plot (Fua et al., 1999). In this figure (as in all the figures hereafter) all values are normalised by dividing them by the maximum value in their respective category, e.g. overlapping value of 0.4 results from 20% over 50%. This type of plot allows values of all combinations to be plotted in one figure and, hence, provides a good overview of how multiple output variables depend on multiple input variable values. Figure 3 (left) shows that the majority of simulations yield duration mean values between 0.46 and 0.75, and that cost mean values are concentrated in the interval (0.77, 0.93). The lines within these duration and cost mean value intervals also indicate that there is a trade-off between duration and cost.

In order to find a combination of input variable values, which would yield target outcomes that are robust to task duration uncertainty modelled in DSM, 'Duration variants' and 'Cost variants' are computed to 'Duration variance' and 'Cost variance' (Figure 3, right). These values represent statistical variances of all simulated duration and cost outcomes. By specifying acceptable upper limits for both duration and cost values, it is possible to identify input variable combinations that can meet the specified range of process performance. In this example the duration mean is set to be less than 0.6 and the cost mean to be less than 0.83 (rectangles in Figure 3, right). This filtration yield seven combinations of input variable values (see Figure 4, left), which then undergo a variance filtration to search for the more robust combinations. In this example, the maximum duration variance is set to be 0.1 and the maximum cost variance to be 0.15 (rectangles in Figure 4, left). Only three combinations remain (Figure 4, right) that can meet this specification, whereas the line labelled with 'RC' seems to offer the best outcome.

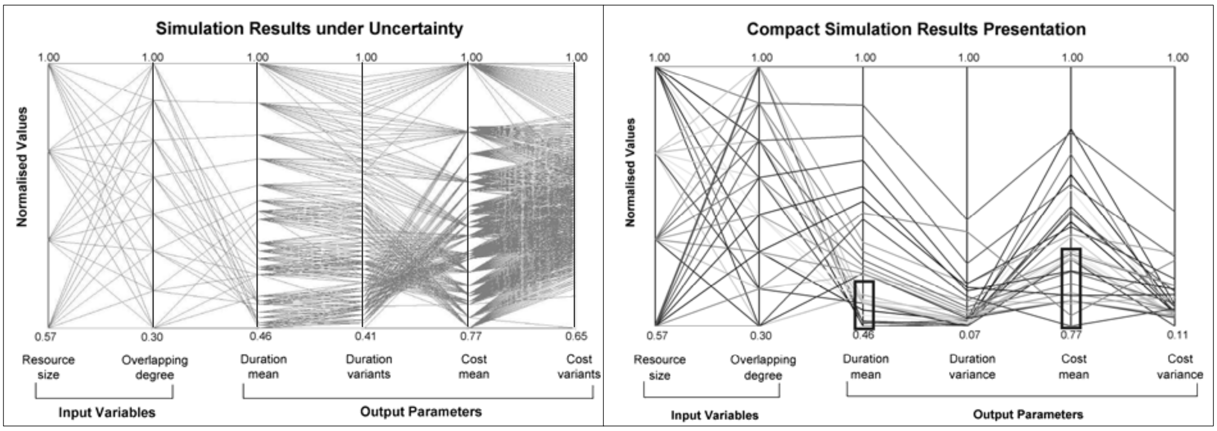


Figure 3. Aggregating simulation results

To further contrast the robustness of the input variable value combination labelled with 'RC' (see Figure 5, left) it is compared to a different combination of input variable values, with the resource size being 0.86 and the overlapping degree being 1 (Figure 5, right). This combination yields similar duration and cost mean output values. However, the ranges of duration and cost variants – caused by task duration uncertainty modelled in the DSM – are much greater than those of the robust combination. Thus, the robust combination poses the better choice in an uncertain project environment.

Despite the simplicity of this example, the case study has shown how the integrated simulation and analysis framework can be used to capture influencing factors from different process abstraction levels – the degree of task overlapping (related to process architecture) and reduced efficiency of resource due to work overload (related to project environmental issues) – and analyse their impact on process performance. By filtering the simulation results it is possible to narrow down the range of alternative management actions and, hence, to provide support to managers in decision making. Furthermore, with the incorporation of task duration uncertainty, the robustness of process performance can be explored, adding more value to insights derived from simulation results.

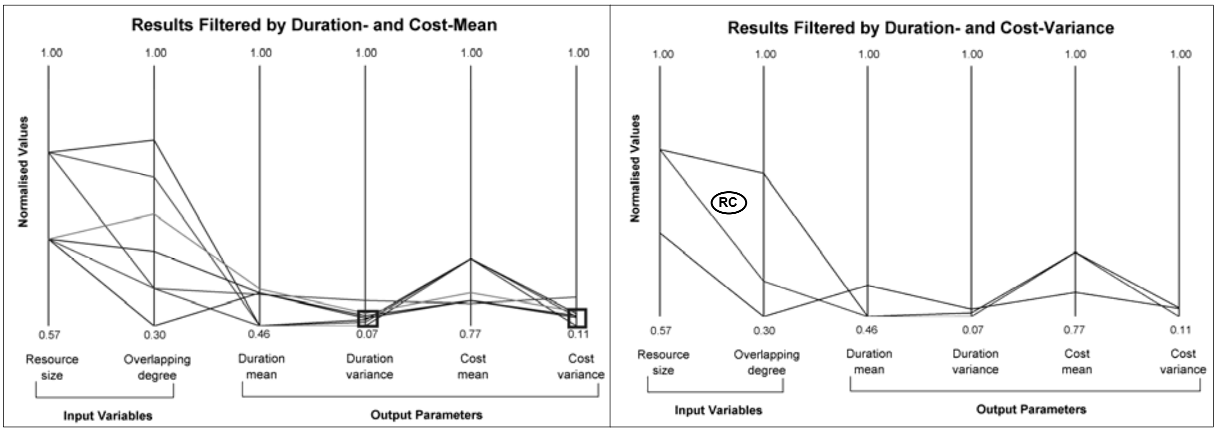


Figure 4. Finding robust strategy

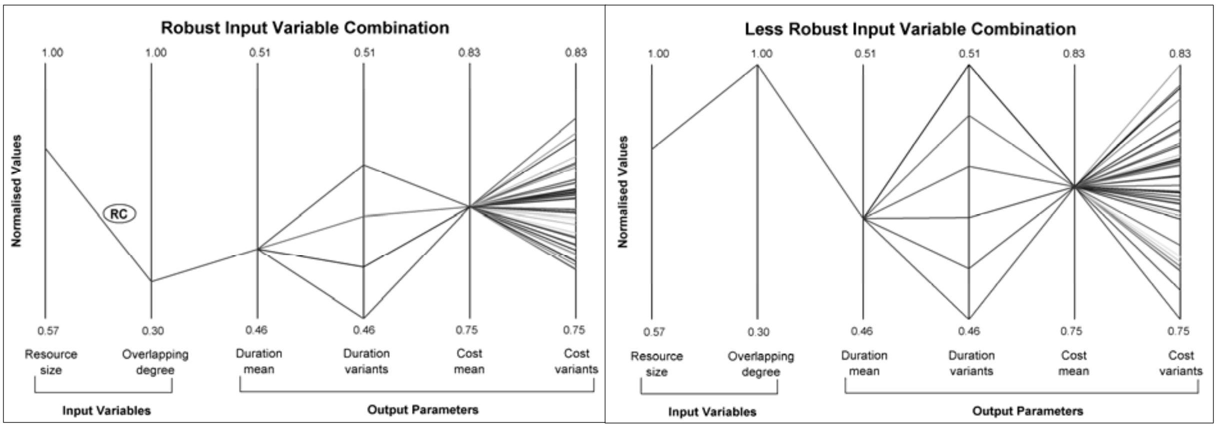


Figure 5. Robustness comparison

4 SUMMARY & OUTLOOK

The initial results of the integrated simulation and analysis method presented in this paper have reflected its potential to evaluate impacts of factors from different abstraction levels on process behaviour within a single framework. Also, by taking task duration uncertainty into account it is possible to investigate the robustness of process performance to duration variabilities.

Further work needs to be undertaken in order to account for more influencing factors and make this method useful for practitioners. In terms of duration uncertainty it would be interesting to investigate how the shape of task duration distribution (e.g. normal, left- or right-skewed distribution) may favour different management actions or policies. Apart from task duration, uncertainty in iteration can significantly influence process behaviour and will need to be considered as well. Furthermore, Pareto-front analysis can be applied to exploit further insights from simulation results. In order to carry out such sophisticated analysis, the model transformation has to be executed fully automatically within a modelling platform (i.e. Cambridge Advanced Modeller) which is in the process of implementation to date. Achieving these steps can make such a method useful for managers.

ACKNOWLEDGEMENT

This research is funded by the Engineering and Physical Sciences Research Council and the Cambridge European Trust.

REFERENCES

Browning, T.R., Fricke, E., & Negele, H. (2005). Key Concepts in Modeling Product development Processes. *Systems Engineering*. Vol. 9, pp. 104-28.

Carracosa, M., Eppinger, S.D., & Whitney, D.E. (1998). Using the Design Structure Matrix to Estimate Product Development Time. *ASME Design Automation Conference*. Atlanta, GA.

Cooper, K.G., Lyneis, J.M., & Bryant, B.J. (2002). Learning to Learn, from Past to Future. *International Journal of Project Management*, Vol. 20, pp. 213-219.

Ford, D.N., & Sterman, J.D. (1998). Dynamic Modeling of Product Development Processes. *System Dynamics Review,* Vol. 14, pp. 31-68.

Fua, Y. H., Ward, M. O., & Rundensteiner, E.A. (1999). Hierarchical Parallel Coordinates for Exploration of Large Datasets. *Conference on Visualization 1999*, San Francisco, California, USA.

Le, H.N., Wynn, D.C., & Clarkson, P.J. (2010). Re-Designing PD Process Architecture by Transforming Task Network Models into System Dynamics Models. *11th International Design Conference – DESIGN 2010*, Dubrovnik, Croatia.

Lyneis, J.M., & Ford, D.N. (2007). System Dynamics Applied to Project Management: a Survey, Assessment, and Directions for Future Research. *System Dynamics Review*, Vol. 23, pp. 157-189.

Contact: H. Nam Le
Engineering Design Centre
University of Cambridge
Trumpington Street
Cambridge CB2 1PZ
United Kingdom
+44 1223 748562
hnl24@cam.ac.uk

BY MODELLING DEPENDENCIES MANAGING COMPLEXITY

Modelling the Influence of Uncertainty, Process Architecture and Feedback Dynamics on PD Projects

H. Nam Le
David C. Wynn
P. John Clarkson

University of Cambridge

Technische Universität München

UNIVERSITY OF CAMBRIDGE

BY MODELLING DEPENDENCIES MANAGING COMPLEXITY

UNIVERSITY OF CAMBRIDGE

Index

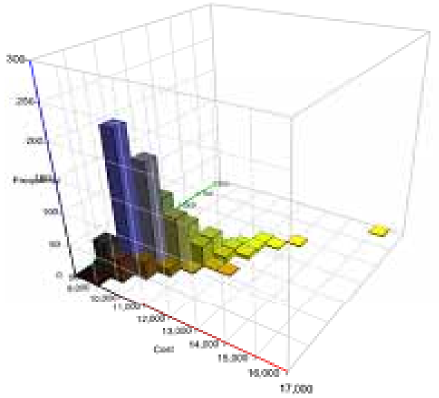
Source: GE Rolls-Royce Fighter Engine Team F136 turbofan engine

- Introduction
- Modelling approaches
- Research questions
- Integrated framework
- Case study
- Summary

Technische Universität München

UNIVERSITY OF CAMBRIDGE

Introduction

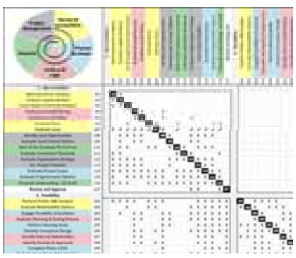
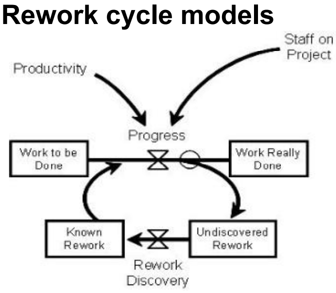


Source: http://www-edc.eng.cam.ac.uk/cam/

- Factors influencing project lead time can be related to:
  - Process uncertainties (e.g. task duration, iteration occurrence)
  - Process architecture (e.g. activity overlapping)
  - Management policies (e.g. workload, schedule pressure)
  - Etc.
- Modelling and simulation can be applied to explore:
  - Impact of constraints
  - Sensitivity to variabilities/uncertainties
  - Direct and indirect impact of management policies
  - Etc.

UNIVERSITY OF CAMBRIDGE

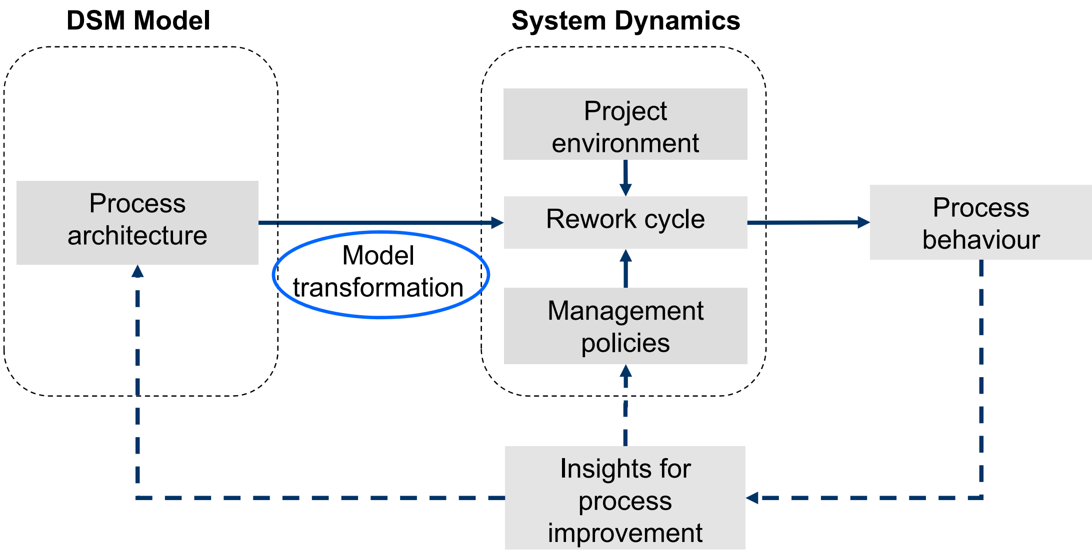
DSM – SD Comparison

	DSM models	Rework cycle models
	 Source: http://web.mit.edu/	 Source: Cooper's rework cycle
Viewpoint	• Network of tasks in a matrix representation	• Phases composed of stocks and flows containing work packages
Key elements	• Dependency relationships • Constraints & Uncertainties	• Rework cycle (stocks & flows) • Process concurrence relationship • Feed-forward and feedback loops
Purposes/ Strengths	• Process architecture • Iteration loops	• Process behaviour analysis • Management policies

Research questions

- Can an integrated modelling framework help to design robust process architectures and optimise lead time?
- How can such a framework be developed?

Framework (1/3) - Overview

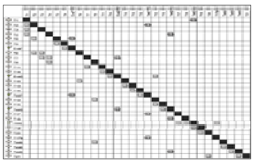


Framework (2/3) – Model transformation

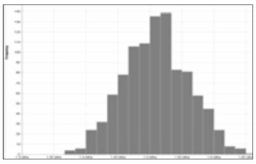
DSM model elements		SD model elements
Process architecture	• Sub-processes	• Rework cycles
	• Task dependency relationships	• Process concurrence relationships
Process details	• Activity duration (uncertainty)	
	• Resource requirement	*under development*
	• Iteration & Rework behaviour	*under development*

Framework (3/3) – Process concurrence relationships

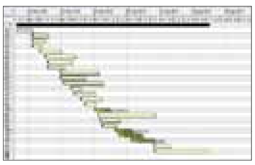
DSM Model



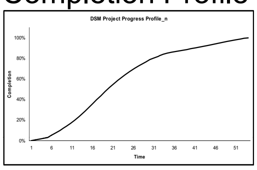
Simulation



Schedule



Completion Profile



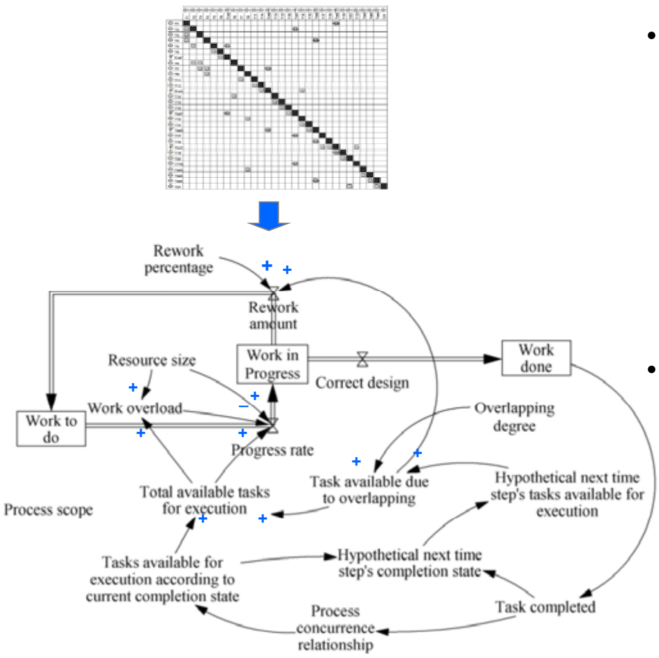
- DSM model with
  - Task dependency relationships
  - Task duration as triangle-pdf
- Set of simulated project lead times with specific schedules
- Process completion profile based on start and end dates of each task
- One time unit of DSM task duration = one work package in SD

Process concurrence relationships

BY MODELLING DEPENDENCIES MANAGING COMPLEXITY

UNIVERSITY OF CAMBRIDGE

Case study (1/3) – Building the model

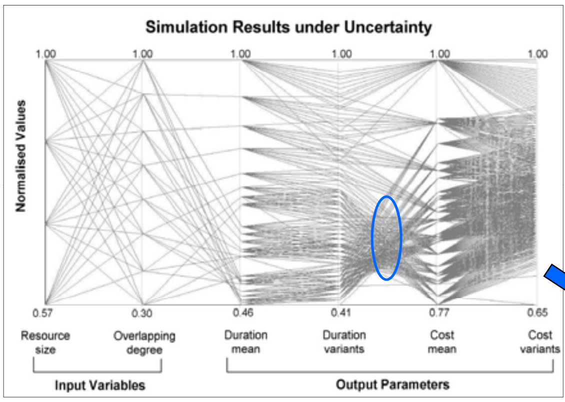


- Model transformation:
  - Hypothetical model with PDF durations
  - Baseline case: no iteration
  - Monte-Carlo simulation
- Extending the rework cycle:
  - Auxiliary variables and key feedback systems (arrows)
  - Assumptions for feedback system influences ("+" and "-" signs)

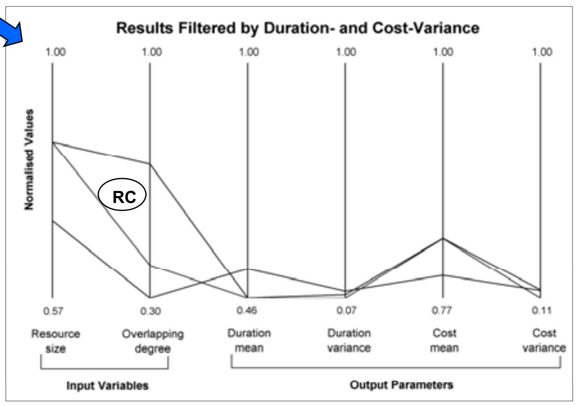
BY MODELLING DEPENDENCIES MANAGING COMPLEXITY

UNIVERSITY OF CAMBRIDGE

Case study (2/3) – Analysis

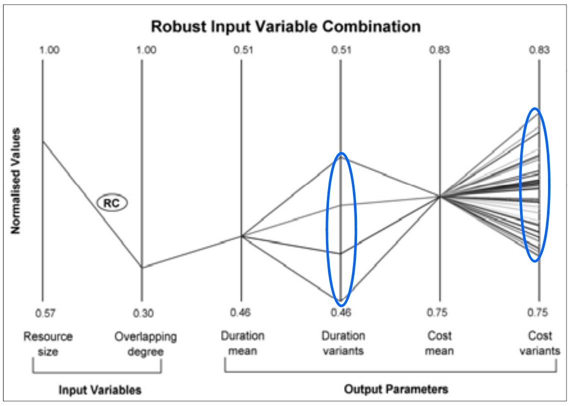


- Task duration uncertainty results in performance uncertainty
- Trade-off between project duration and cost (circled)

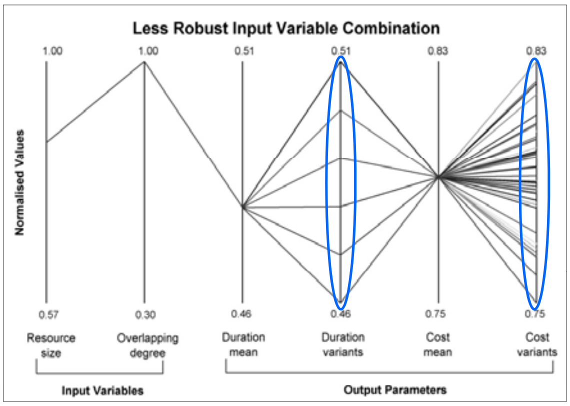


- Filtering out acceptable values of mean output parametes
- Searching for robust process with low variance values

Case study (3/3) – Finding robust strategy



- Similar mean values
- Less sensitive to task duation uncertainty due to smaller ranges of output parameter variants



Summary and outlook

- Framework can encompass influencing factors addressed by different modelling approaches
- Provides insights into process robustness in the light of task duration uncertainty
- Further work will include
  - Comparison of integrated framework simulation results with task network model's results
  - Theory development (e.g. iteration uncertainties)
  - Tool development (i.e. implementation in the Cambridge Advanced Modeller)
  - Exploration of scenarios of application (e.g., activity overlapping with iteration, frequency of design review)
  - More sophisticated analysis (e.g. sensitivity of lead time to task duration uncertainty)
  - Etc.