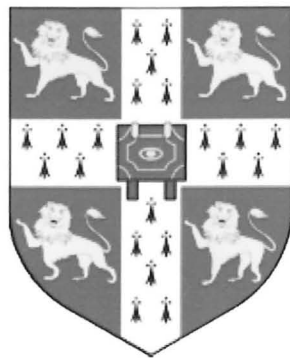


PhD. 29953

Supporting Design Planning Through Process Model Simulation



Tomás Flanagan

Churchill College



May 2006

*A dissertation submitted for the Degree of Doctor of Philosophy
Cambridge University Engineering Department*

Declaration

Except where otherwise stated, this thesis is the result of my own research and does not include the outcome of work done in collaboration.

This thesis has not been submitted in whole or in part for consideration for any other degree qualification at this or another university.

The thesis contains 94 figures, 26 tables and less than 65,000 words.

Tomás Flanagan

Tomás Flanagan

Churchill College

Cambridge

May, 2006

Abstract

Modelling and analysis of design processes is non-trivial and predicting the behaviour of such systems is especially challenging. Nonetheless, the effective planning of design projects is critical for many engineering companies.

Based on the findings from a thorough literature review and an extensive industrial case study, this thesis identified four themes on which subsequent process analyses focused: modelling and representation; scale and connectivity; rework, and the product-process link. The case study also provided a deeper understanding of the practical challenges of planning in industry.

In order to meet these challenges, both the Signposting modelling framework and simulation tool were enhanced to provide increased functionality for process modelling, representation and analysis. Further, novel approaches for process analysis – the use of hypothetical models and confidence profiles that link product and process information – were proposed. A software tool was developed to automatically generate such hypothetical models.

Simulation analyses were performed on both hypothetical and real-world models. The results elucidated the effects of structural variations in terms of scale and connectivity on project performance. They also showed how task reordering due to rework can lead to major process delays, even when the time taken to rework failed tasks is extremely short. Further, the simulation analyses demonstrated how confidence profiles could be used to identify rework early in the process and reduce the resulting project schedule impact. The results from the simulation analyses were evaluated against historical data from the case study company and heuristics for design planning were defined.

Overall, this research demonstrated how simulation analysis of both real-world and hypothetical models, using the enhanced Signposting tool, can provide useful insights into design process planning.

Acknowledgements

I would like to thank Dr. Claudia Eckert for supervising this research: in addition to excellent advice and support from an academic perspective, her constant encouragement throughout the research period was invaluable. She went above and beyond what can reasonably be expected of a supervisor. I also wish to express deep gratitude to Prof. John Clarkson, both for acting as an academic advisor and for partially funding this research. Additional funding was provided by the Cambridge European Trust, the Isaac Newton Fund and Churchill College.

Several other academics also contributed to the thesis. Conversations with Dr. Rob Bracewell, Dr. Dave Brown, Dr. Martin Stacey, Prof. Larry Bucciarelli and Prof. Dan Whitney reminded me of what Feynman termed 'the pleasure of finding things out'. I would also like to thank all members of the EDC, especially the DPI group, Nathan and Andrew. I'm especially grateful to René and Seena for their contributions to developing the Signposting tool, to Dave and Labi for useful discussions, and to Anja and Tido for proofreading the thesis.

From an industrial perspective, the support of Perkins engines was instrumental in shaping this research. Particular acknowledgement is due to Dr. Donald Campbell, Simon Yarram, Chris Gordon, Richard Mariniak and Tony Robinson.

Many other people have contributed to my thesis indirectly. I shall be forever indebted to my friends who made my time in Cambridge so enjoyable: notably, the Churchill dinner group (Hyun-Jin, Islam, Rasmus, Sam, Sig, Thierry, Yuri and Zdenek), my housemates (Becky, Helen, Mike, Richard, Robin and Yen), the boat club, the volleyball team and my MCR football team. Finally, I would like to thank the Irish crew, especially my family.

Contents

1	INTRODUCTION	1
1.1	PROBLEM DEFINITION	2
1.1.1	<i>The planning challenge</i>	4
1.1.2	<i>The search for robust processes</i>	4
1.2	RESEARCH MOTIVATION	6
1.2.1	<i>Motivation from a design research perspective.....</i>	7
1.2.2	<i>Motivation based on previous case studies</i>	9
1.3	KEY RESEARCH QUESTIONS.....	10
1.4	THESIS STRUCTURE.....	12
1.5	SUMMARY.....	13
2	METHODOLOGY	14
2.1	OVERALL RESEARCH METHODOLOGY	15
2.1.1	<i>Choosing a methodological framework.....</i>	15
2.1.2	<i>Linking thesis-structure to methodology</i>	17
2.2	CASE STUDY.....	19
2.2.1	<i>Interviews</i>	19
2.2.2	<i>Observational studies</i>	24
2.2.3	<i>Documentation</i>	27
2.2.4	<i>Summer student collaboration.....</i>	27
2.3	THEORY AND TOOL DEVELOPMENT	28
2.3.1	<i>Building on existing research.....</i>	28
2.3.2	<i>Theory development.....</i>	29
2.3.3	<i>Development of tools.....</i>	30
2.3.4	<i>Application in context.....</i>	31
2.4	SUMMARY.....	32
3	RESEARCH INTO PLANNING AND MODELLING COMPLEX DESIGN PROJECTS	33
3.1	COMPLEXITY.....	34
3.1.1	<i>Non-linear behaviour and bifurcation points.....</i>	36
3.1.2	<i>Attractors and emergence.....</i>	38
3.1.3	<i>Complex adaptive systems.....</i>	39
3.1.4	<i>Information, entropy and complex system resetting</i>	39
3.1.5	<i>Relating different views on complexity</i>	40
3.2	THE COMPLEXITY OF DESIGN PROCESSES	40
3.2.1	<i>Generic models of design processes.....</i>	41
3.2.2	<i>Characteristics of design processes which lead to planning challenges.....</i>	43
3.2.3	<i>Design planning: a complex-system behaviour-prediction problem</i>	47

3.3	APPROACHES TO ANALYSING AND PLANNING COMPLEX PROCESSES	48
3.3.1	<i>Simulation analysis</i>	49
3.3.2	<i>Operations research</i>	51
3.3.3	<i>AI planning</i>	55
3.4	MODELS USED TO PLAN DESIGN PROCESSES	59
3.4.1	<i>DSM</i>	60
3.4.2	<i>IDEF</i>	61
3.4.3	<i>Signposting</i>	63
3.4.4	<i>Other modelling frameworks</i>	65
3.4.5	<i>Comparing the different models</i>	66
3.5	CHAPTER REFLECTION: OPPORTUNITIES TO SUPPORT DESIGN PLANNING.....	68
4	PROJECT PLANNING: AN INDUSTRIAL CASE STUDY	70
4.1	PERKINS ENGINES: COMPANY DESCRIPTION	71
4.2	CHARACTERISATION OF PERKINS DESIGN CONTEXT	72
4.2.1	<i>Increasing product complexity</i>	72
4.2.2	<i>New process risks</i>	73
4.2.3	<i>Less experienced people</i>	74
4.2.4	<i>Six-sigma culture</i>	75
4.3	PLANNING PRACTICE IN PERKINS	75
4.3.1	<i>Multiple types of plans</i>	76
4.3.2	<i>Links between plans</i>	81
4.3.3	<i>Uses of plans</i>	83
4.3.4	<i>Creating and owning plans</i>	86
4.4	PROJECT PLANNING CHALLENGES IN PERKINS	88
4.4.1	<i>Modelling uncertainty and contingency</i>	88
4.4.2	<i>Scale and connectivity</i>	92
4.4.3	<i>Iteration and rework</i>	94
4.4.4	<i>Confidence-driven planning</i>	95
4.4.5	<i>Coping with imperfect plans: the importance of overview</i>	98
4.5	COMPARISON WITH OTHER DESIGN CASE STUDIES.....	101
4.5.1	<i>Observations from EDC case studies</i>	101
4.5.2	<i>Studies outside the EDC</i>	103
4.6	CHAPTER REFLECTION: OPPORTUNITIES TO SUPPORT DESIGN PLANNING.....	105
5	FOUNDATIONS FOR ROBUST PLANS	107
5.1	IMPROVED MODELLING OF COMPLEX DESIGN PROCESSES.....	108
5.1.1	<i>Model building and evaluation</i>	108
5.1.2	<i>Representing uncertainty</i>	109
5.1.3	<i>Enhancing the Signposting model structure</i>	110
5.1.4	<i>Separating model analysis from process analysis</i>	112
5.1.5	<i>Requirements concerning modelling and representation</i>	113

5.2	INVESTIGATING THE EFFECTS OF PROCESS STRUCTURE	113
5.2.1	<i>Exploring the effects of process-specific variations</i>	<i>113</i>
5.2.2	<i>Exploring the effects of structural variation.....</i>	<i>115</i>
5.2.3	<i>Requirements concerning model variations</i>	<i>116</i>
5.3	REWORK IMPACT MITIGATION.....	116
5.3.1	<i>Accounting for likely rework</i>	<i>116</i>
5.3.2	<i>Understanding process-sensitivity to task-failures.....</i>	<i>117</i>
5.3.3	<i>Requirements concerning rework.....</i>	<i>118</i>
5.4	EXPLORATION OF THE PRODUCT-PROCESS LINK.....	119
5.4.1	<i>Signposting confidence versus Perkins confidence</i>	<i>119</i>
5.4.2	<i>Confidence-based process-analysis.....</i>	<i>120</i>
5.4.3	<i>Confidence and rework.....</i>	<i>122</i>
5.4.4	<i>Requirements concerning the product-process link.....</i>	<i>122</i>
5.5	SUMMARY.....	123
6	DEVELOPING A TOOL FOR DESIGN PROCESS ANALYSIS	124
6.1	A SOLUTION PROPOSAL FOR DESIGN PROCESS ANALYSIS.....	125
6.1.1	<i>Specific requirements for concept embodiment.....</i>	<i>126</i>
6.2	ANALYSIS OF HYPOTHETICAL DESIGN PROCESS MODELS.....	129
6.2.1	<i>Advantages of hypothetical models</i>	<i>129</i>
6.3	BUILDING REAL-WORLD PROCESS MODELS	130
6.3.1	<i>Modelling task precedences in Signposting.....</i>	<i>131</i>
6.3.2	<i>Modelling parallel tasks using Signposting.....</i>	<i>132</i>
6.3.3	<i>Software support for modelling.....</i>	<i>133</i>
6.4	INPUT-MODEL VISUALISATION AND ANALYSIS	135
6.4.1	<i>Gantt chart and hierarchical Gantt chart</i>	<i>135</i>
6.4.2	<i>Design Structure Matrix and Domain Mapping Matrix</i>	<i>136</i>
6.4.3	<i>Confidence profiles.....</i>	<i>136</i>
6.5	MODEL PERTURBATION CAPABILITIES	137
6.6	ENHANCED SIMULATION CAPABILITIES	138
6.6.1	<i>Overview of the simulation code</i>	<i>138</i>
6.6.2	<i>Enhanced simulation capabilities.....</i>	<i>139</i>
6.6.3	<i>Improved algorithms for picking tasks</i>	<i>140</i>
6.7	OUTPUT VISUALISATION AND ANALYSIS	141
6.7.1	<i>Representative process runs</i>	<i>142</i>
6.7.2	<i>Histograms and scatter plots.....</i>	<i>142</i>
6.7.3	<i>Gantt chart subject to uncertainty.....</i>	<i>143</i>
6.7.4	<i>Design Structure Matrices.....</i>	<i>144</i>
6.7.5	<i>Confidence-based simulation analysis</i>	<i>144</i>
6.7.6	<i>Multiple views.....</i>	<i>145</i>
6.8	VERIFICATION AGAINST REQUIREMENTS.....	146
6.9	SUMMARY.....	148

7	LESSONS FROM SIMULATION OF HYPOTHETICAL MODELS	149
7.1	GENERATING HYPOTHETICAL MODELS	150
7.1.1	<i>Generation of hypothetical models</i>	<i>150</i>
7.1.2	<i>Implementation of the model generator</i>	<i>152</i>
7.1.3	<i>Insights from the software development process</i>	<i>152</i>
7.1.4	<i>Limitations of hypothetical, generated models</i>	<i>154</i>
7.2	VARIATIONS DUE TO SCALE AND CONNECTIVITY	154
7.2.1	<i>Process model variations due to scale</i>	<i>154</i>
7.2.2	<i>Process variation due to connectivity-level</i>	<i>159</i>
7.3	VARIATIONS DUE TO REWORK	163
7.3.1	<i>Variations due to task acceleration during rework</i>	<i>163</i>
7.3.2	<i>Sensitivity to rework-likelihood values</i>	<i>166</i>
7.3.3	<i>Insights into rework</i>	<i>169</i>
7.4	VARIATIONS IN CONFIDENCE	169
7.4.1	<i>Variations due to the timing of confidence-growth</i>	<i>169</i>
7.4.2	<i>Confidence variations subject to tight deadlines</i>	<i>171</i>
7.5	EVALUATION OF RESULTS AGAINST REAL-WORLD DATA	172
7.5.1	<i>Scale and connectivity</i>	<i>173</i>
7.5.2	<i>Rework</i>	<i>173</i>
7.5.3	<i>Confidence</i>	<i>175</i>
7.6	IMPLICATIONS OF RESULTS: HEURISTICS FOR PLANNING	175
7.6.1	<i>Scale and connectivity</i>	<i>175</i>
7.6.2	<i>Rework</i>	<i>177</i>
7.6.3	<i>Confidence</i>	<i>178</i>
7.7	SUMMARY	178
8	LESSONS FROM SIMULATION OF AN INDUSTRIAL PROJECT	180
8.1	MODEL ELICITATION	181
8.1.1	<i>Building the Signposting model</i>	<i>181</i>
8.1.2	<i>Model description</i>	<i>182</i>
8.1.3	<i>Insights from building the process model</i>	<i>184</i>
8.1.4	<i>Insights from analysis of the simulation input model</i>	<i>186</i>
8.2	VARIATIONS DUE TO REWORK	186
8.2.1	<i>Rework probabilities</i>	<i>187</i>
8.2.2	<i>Rework duration variation</i>	<i>190</i>
8.3	VARIATIONS DUE TO CONFIDENCE: A PRODUCT-PROCESS PERSPECTIVE	191
8.3.1	<i>Confidence and task-prioritisation</i>	<i>191</i>
8.3.2	<i>Confidence and rework</i>	<i>193</i>

8.4	EVALUATION OF SIMULATION RESULTS.....	195
8.4.1	<i>Rework</i>	195
8.4.2	<i>Confidence</i>	197
8.4.3	<i>Limitations</i>	198
8.5	IMPLICATIONS OF RESULTS: HEURISTICS FOR PLANNING	198
8.5.1	<i>Modelling and representation</i>	198
8.5.2	<i>Rework</i>	200
8.5.3	<i>The product-process link through confidence</i>	201
8.6	CHAPTER SUMMARY	202
9	CONCLUSIONS	203
9.1	CORE CONCLUSIONS.....	204
9.2	RELATED RESEARCH CONTRIBUTIONS.....	204
9.3	CONTRIBUTIONS TO PERKINS	205
9.4	EVALUATION AGAINST RESEARCH OBJECTIVES	206
9.5	LIMITATIONS OF THIS WORK	210
9.6	OPPORTUNITIES FOR FUTURE WORK	211
9.7	SUMMARY.....	215
10	APPENDIX.....	217
	EXAMPLE HYPOTHETICAL MODEL	217
	SIMPLIFIED EXAMPLE MODEL (TO DEMONSTRATE SIMULATION)	219
	BIBLIOGRAPHY	223

1

Introduction

On August 10th 1628, the Vasa – a Swedish warship – sank ten minutes into her maiden voyage. Over thirty people drowned as a direct consequence; the indirect impact on Swedish naval history is incalculable (Fairley and Willshire, 2003).

The disaster took place due to a combination of project mismanagement issues which resulted in a design failure. The project was performed subject to considerable time-pressure: Sweden was at war with Poland and urgently needed the new ship – the loss of 10 Swedish naval ships during a storm added further time-pressure.

The resulting challenges were exacerbated by the Swedish king, who changed the ship's requirements late in the project. On March 20th, 1628 (less than 5 months before the tragedy) Hybertson, the shipbuilder, reports that a ship is under construction and that its keel length is 120 feet, although archaeological and historical evidence suggests that the Vasa began as a 111 foot vessel and was extended to 136 feet after a meeting between the king and the shipbuilder on March 21st. It seems that the design of the ship changed significantly less than 6 months before her launch.

As the project progressed, unfavourable results from an important test, which could have prevented the loss of the ship, were ignored. A stability test, which consisted of

30 men running from side to side on the ship's deck, was prematurely ended because the ship was rocking so violently that it looked as if it would capsize. Surprisingly, considering the importance of the test, neither the project manager (the shipbuilder) nor the king were informed of the test-results and no action was taken to make the ship more stable.

Finally, the technical challenges of designing the Vasa were also instrumental in the project's failure: it was larger than contemporary ships and had two gun-decks with heavy artillery when the norm was to place lighter guns on the upper gun-deck. Ultimately, it was this design characteristic which caused the ship to become unstable and sink.

Almost 400 years later, the desire to design better products faster prevails and many of the associated problems of project mismanagement remain unresolved – many projects are “overdue and over-budget, over and over again” (Evans, 2005). Even when budget and schedule targets are met, there is no guarantee that the product meets its quality objectives.

1.1 Problem definition

The fate of the Vasa brings to light many important design issues: Firstly, it illustrates the importance of scale and complexity: knowledge of ship design, based on smaller simpler ships, was used to design the Vasa but this data was incorrectly extrapolated for the design of the larger vessel (Fairley and Willshire, 2003). More recently, failure of the Ariane 5 took place when a scaling factor was overlooked while reusing software code from the smaller, simpler Ariane 4 (Lions, 1996).

Secondly, the Vasa disaster underscores the link between a product and its design process: a stability test uncovered the ship's flaw but there was no time to suitably redesign the ship because of the tight completion deadline imposed by the Swedish king (Hendrickson, 2002). Had more time been available, the design process could have been adapted to improve the product's quality. The decision to ignore the results of a test finds parallels with the Firestone/Ford tyre failures where executives knew about the problem but failed to react appropriately (Greenwald, 2001); they failed to modify the process to address the product weaknesses and suffered costly lawsuits as a result.

Thirdly, the specification for the Vasa was changed by the king, late in the design process, when he requested that the size of the ship be increased from 111 feet to 136 feet, and these changes resulted in major rework (Fairley and Willshire, 2003). Cooper (1993) argues that rework causes projects to run billions of dollars over budget and years behind schedule. For example, in 2003, the National Audit Office found that spending on the UK MoD's top 20 projects was projected at £3.1 billion over budget and an average of 18 months behind schedule (National Audit Office, 2004).

Finally, the Vasa highlights a modelling problem: at the time of its design, ship designers did not know how to construct a simple, abstract model of the ship that could be used to determine stability. As a result, testing was performed based on the final product leading to the late problem discovery, which made corrective action expensive. A more recent example of disaster due to a modelling limitation is the Tacoma Narrows Bridge failure where aerodynamic forces were not adequately considered. While both of the examples concern physical products, modelling challenges also arise for more abstract systems – e.g. stock markets, weather, society and, a key topic for this thesis, design processes.

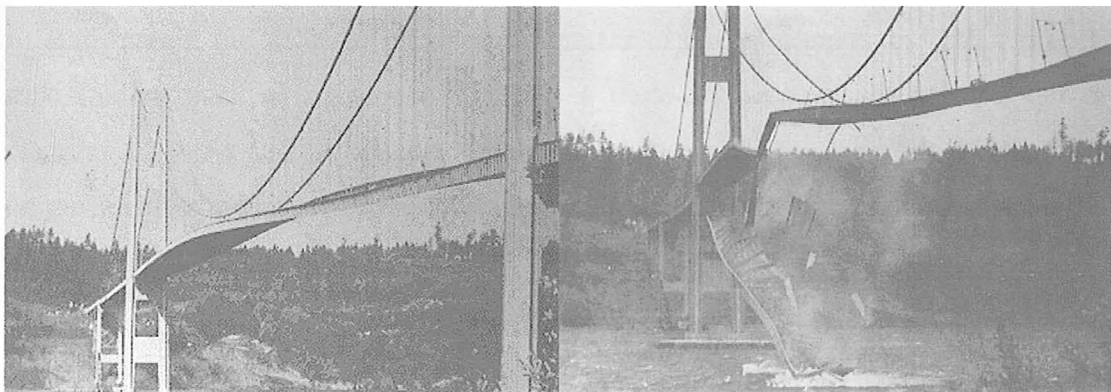


Figure 1.1 The collapse of the Tacoma Narrows Bridge (Irving, 1999)

These issues of 1) scale and complexity, 2) dependencies between product and process, 3) rework and 4) modelling and representation are the core themes which intertwine the chapters of this thesis. They are especially relevant in the current design context, where market pressure demands enhanced products within the constraints of ever-tighter deadlines.

1.1.1 The planning challenge

Effective planning can prove essential to achieving success in large-scale industrial projects. Due to the themes described above, however, design planning frequently proves difficult and error-prone. The design of large-scale, complex products involves thousands of tasks and potentially millions of interdependencies, if indirect task connections are considered. In addition to other applications which will be described later (Chapter 4), plans are used to organise the workload and help ensure that the project is completed on time. However, the scale and complexity of the design process introduces numerous planning challenges.

Commonly used project representations, such as Gant charts, fail to adequately capture 1) process risk due to product-related uncertainties and 2) the impact of task-failures. These limitations are closely aligned to the themes of modelling, rework and the product-process link. Tough time constraints can leave designers with the difficult choice between meeting deadlines and refining the design to the desired level of quality. Examining either the product or the process in isolation results in an incomplete picture. Planning challenges also arise when tasks fail to produce the expected information, thus leading to the rework of dependent tasks.

In many cases, the value of planning is unclear to both designers and other process stakeholders, such as managers. There is a trade-off between the effort spent on planning a project and the rewards obtained: in some cases the expert decisions based on gut-feelings are largely correct and the time saved on project planning justifies the associated risks.

In order to elucidate the benefits of planning, and to improve the way in which projects are planned, it is necessary to better understand how different process characteristics – scale, connectivity level, rework likelihood, task duration during rework – affect project performance. Such knowledge is also helpful in increasing process robustness.

1.1.2 The search for robust processes

The actual process which unfolds during the design of a new product is almost inevitably different from the most likely process route envisaged during the initial planning phase. During process execution, new requirements emerge, tasks take longer than expected, problems with the design occur and unexpected test-results are

obtained. Hence, the actual day-to-day work of the designer can diverge considerably from that outlined in the project plan. Although problems also arise during manufacturing, the wide variety of design uncertainties makes design work especially difficult to plan.

Despite this potential for divergence, it is not possible to re-plan a project every time something changes. Hence, plans should cover a host of eventualities not just a chain of most likely task-outcomes. Companies need robust plans which remain useful even when unexpected events unfold. To this end, different process risks must be identified and steps taken to minimise their impact on the process, thus increasing its robustness. The word robust can be defined in terms of the ability to yield approximately correct results despite the falsity of certain assumptions or inaccuracy of certain parts of the input (Simpson and Weiner, 1989). In this light, robust processes are processes which are insensitive to different uncertainties (rework, task-delays, resource fluctuations etc). Currently however, the characteristics of robust processes are ill-defined.

Many design process improvement initiatives focus on reducing new-product development-time. Process models are built at the beginning of a design process and analyses undertaken to predict how the process will behave. Managers estimate task dependencies and uncertainties to get a feeling for the likely duration and risk associated with the project but the combined effect of different uncertainties is difficult to predict. Because analyses are based on estimated values, a range of possible outcomes must be considered.

Simulation analysis of process models (Chapters 3, 6, 7, 8) can be used to determine the frequency with which different outcomes are likely to occur. It can show how different task-failures or delays, as well as resource shortfalls are likely to affect the process, thus providing useful insights. Assuming that accurate models are used for the simulation analysis, reducing the mean of the project duration from the different simulation runs equates to shortening the development time of the actual project. In contrast, increasing process robustness corresponds to reducing the variance associated with different process execution scenarios (Fig. 1.2).

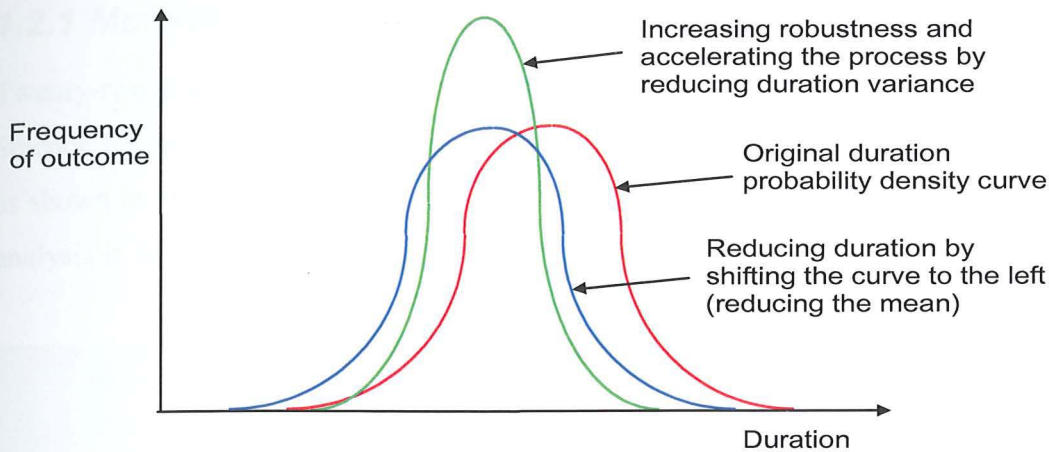


Figure 1.2 Increasing process robustness

In the above diagram (Fig. 1.2), the red curve describes the initial process duration while the blue curve describes the result of process improvements which reduce the mean expected process duration without affecting the predicted variation. In contrast the green curve describes a more robust process which is less variable in terms of duration.

Some changes to the design process have little effect, while others have a major impact. To improve processes, it is necessary to understand where interference will have the maximum benefit and which changes to avoid, if possible. Knowledge of these process levers and pitfalls is critical in reducing product development times and in defining the characteristics of robust processes (Chapters 7 and 8). Simulation is a powerful tool for their identification.

Simulation analysis of real-world processes is a useful approach for exploring sensitivity to different uncertainties. However, process-specific analyses fail to answer questions such as “how does scale and connectivity-level affect process robustness?” and “how does learning during rework influence robustness?” In addition, fixation on real-world models can limit the range of analyses considered. This thesis will discuss how the analysis of hypothetical models, in complement to real-world models, can address these challenges.

1.2 Research motivation

The motivation for this work, within the overall context of engineering design, is argued below from a literature viewpoint and based on industrial case studies.

1.2.1 Motivation from a design research perspective

Twenty-two academic and industrial experts at a workshop supported by the National Science Foundation (NSF) of the USA (Shah, 1996) prioritised the needs of industry as shown in Table 1.1. The work described in this thesis considers the role of process analysis in meeting these objectives.

No.	Industry Need Importance	Importance (scale 1-10)
P1	Improve quality of design	9
P2	Facilitate team decision-making	9
P3	Improve design environments	8
P4	Create seamless integration between design and analysis	7
P5	Understand company's product realisation process	7
P6	Archive and reuse design history	6
P7	Determine impact of decisions	6

Table 1.1 Industrial needs

Improving understanding of the manner in which process characteristics and uncertainties affect project performance lies at the heart of this research. This objective aligns closely with the goal of understanding companies' product realisation processes (P5).

The realisation of more robust processes can improve efficiency and remove errors that result in unnecessary design effort and costly changes. Opportunities exist to translate these benefits into improved product quality by assigning extra resources to key tasks or by performing additional iterations to refine the solution (P1). Analysis of the product-process link is a core topic of this thesis.

Many of the factors shown in the above table are interlinked: for example the goal of determining the impact of decisions (P7) relates to facilitating team decision-making (P2) and may indirectly improve the design quality (P1). Understanding the company's product realisation process (P5) is important during decision making; by tackling this issue directly, this thesis will also contribute to meeting the other industrial needs outlined above.

Wallace and Blessing (1999) have elaborated on the industrial needs identified in Table 1.1 stating that *“products and processes are getting more complicated and design teams need research to provide theories, methods and tools to help them manage and integrate this complexity.”* They predict increased research in the following areas: 1) the complexity of processes; 2) system-based knowledge and 3) systems rather than products. This research fits these criteria.

The need to design better products faster

Several researchers have noted the importance of reducing time-to-market in determining product success (DTI, 1992; Ruppert, 1994; Jarrett, 2000; Jarrett and Clarkson, 2001). Ruppert's work discusses how producing the Boeing 747 airplane in three years when the industrial norm was five gained the company a strong advantage over competitors. A study by McKinsey for the UK Department of Trade and Industry (DTI, 1992) has shown that if a project is completed six months late, the percentage loss in after-tax profit will be 30% while exceeding production costs by 9% will lead to a profit loss of about 20%. Surprisingly, if a project finishes on time but overruns the development cost by 50% this leads to a profit loss of less than 5%. While the generality of these results for all types of products and market contexts is unproven, they nonetheless illustrate the need to reduce new product development times.

Another example, which illustrates the need to design better products faster, is that of diesel-engines, which are the subject of the case study in this thesis: product-innovations are driven by emissions legislation and a product which fails to meet these regulations cannot be sold (Jarratt et al., 2003). Also, emissions legislation limits the time available for new product development programmes, requiring the accelerated design of superior products.

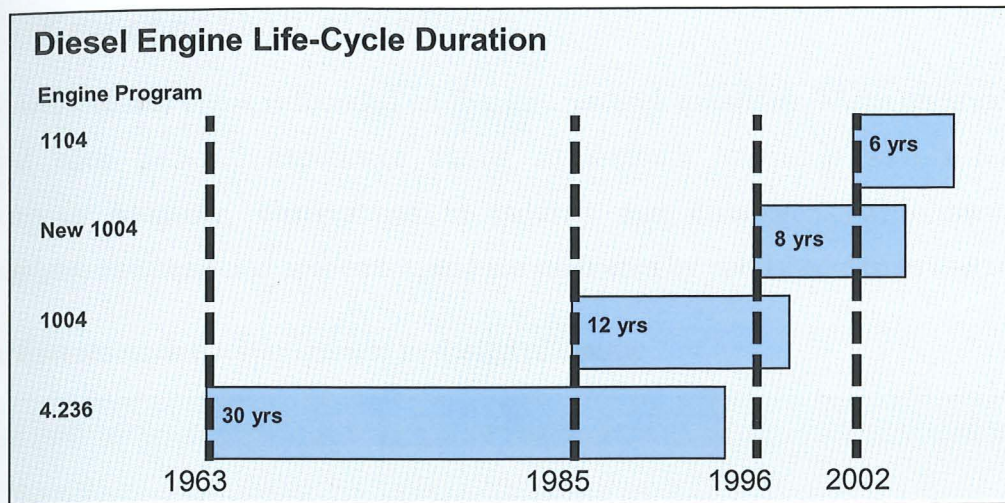


Figure 1.3 Decreasing product development times: diesel engine example (Jarratt et al., 2003)

1.2.2 Motivation based on previous case studies

This research is also motivated by several industrial case studies, performed by members of the EDC, each involving approximately twenty one-hour interviews with designers and design managers (Eckert and Clarkson, 2003; Flanagan et al., 2003a; Jarratt et al., 2003). These interviews were complemented by observations within the companies and feedback from the participants on issues identified during analysis of interview recordings.

The unifying theme of these studies was process improvement but communication, engineering change and project planning were also considered. The relationship between project planning and communication is described by Eckert and Clarkson (2003). In the consultancy arm of a large engineering firm, distant management and strong personal animosities led to several planning and communication challenges.

A study at a diesel engine manufacturer (the same company as described in Chapter 4 of this thesis) focused on changes to existing products as well as incremental new product development. Although the main focus was on changes to products to meet customisation demands, this study also highlighted the role of change in causing delays to project completion.

Overall the studies showed that problems relating to project planning are commonplace in engineering design companies, and are supported by findings from other case studies in engineering design (Browning, 2001; Yassine et al., 2001). The studies showed that planning problems are multi-causal and that the impact of different factors on project performance is poorly understood.

1.3 Key research questions

The work in this thesis is founded on five key research questions. These questions are based upon personal experience during an 8-month internship with a leading aerospace company, observations in industry and discussions with managers, designers, engineers and academics, and are supported by published design literature.

The first research question relates to design planning:

How are large-scale design projects planned and what are the challenges involved?

This leads to the following questions:

- Who plans projects?
- Who uses plans?
- What form do project plans take?
- How is information from different plans linked together?
- How do plans reflect risk and uncertainty?

The second research question concerns sensitivity to process properties:

(How) Do process properties such as scale, connectivity-level and rework affect plannability?

This sparks several follow-on questions:

- What is the importance of scale on plannability?
- What is the importance of complexity, in terms of task-connectivity level, on plannability?
- How does rework affect plannability?
- How can knowledge of the process structure be utilised to construct more robust plans?

The third research question relates to modelling of the product-process link:

(How) Can modelling of product-process interdependencies improve the quality of plans and reduce process risks?

This inspired the following questions:

- How should interdependencies between product and process be modelled?
- Can modelling the product-process link yield insight into rework behaviour?
- How can process risk be reduced through analysis of the product-process link?
- Can better understanding of the product-process link lead to process improvements and/or better plans?

This thesis will use simulation to explore process sensitivities to scale, connectivity-level and rework, and to explore product-process interdependencies. The fourth research question relates to the role of simulation.

How can simulation be most effectively deployed to determine the impact of different process properties on project performance and plannability?

Questions that arise from this research question are:

- How can simulation be used to support process analysis and project planning?
- What can be learned about design processes by simulating real-world models?
- What can be learned about design processes by simulating hypothetical models?

The fifth research question considers tools and techniques for design project analysis:

How can support tools and techniques, which aim to support engineering-design planning, be improved?

This leads to the following questions:

- What tools are currently used to plan design projects?
- What are the deficiencies of existing tools?
- How do projects succeed despite the limitations of current tools?
- How can improved software tools provide better support for planning?

Broadly speaking, the research described here aimed at understanding how design process characteristics affect process behaviour and hence at improving planning practice. The work set out to create new knowledge in the area of engineering design project planning that would be of benefit to industry and academia. More specifically, it aimed to answer the questions outlined above.

1.4 Thesis structure

In light of the key research questions outlined above, the thesis proceeds as follows: Chapter 2 describes the research methodology for this work. It is followed by an overview of literature on design process modelling, project planning, project management, AI planning, complexity in design and simulation (Chapter 3). Next, Chapter 4 provides a detailed description of project planning in industry, based on an extensive case study performed at Perkins Engines Company Limited. Building on the results of the literature review and the case study, the foundations for robust plans are discussed in Chapter 5, and the research requirements are refined. A tool for increasing process robustness is proposed in Chapter 6. Chapter 7 describes the simulation analysis of hypothetical, generated models and includes a comprehensive investigation of rework in engineering design projects. This is followed by the simulation analysis of a real-world industrial project and a discussion of resultant implications for industry (Chapter 8). Finally, a summary of the key research contributions, along with opportunities for further research, concludes the thesis.

Chapter	Content
Introduction	Motivation
Methodology	Research approach
Literature review	Problem analysis
Case study	
Foundations for robust plans	Requirements definition
A tool for increased process robustness	Solution conceptualisation & implementation
Analysis of hypothetical models	Analysis & evaluation
Analysis of a real-world model	
Conclusions	Contributions and further work

Figure 1.4 Thesis structure

1.5 Summary

Based on findings from a literature review and an industrial case study, this thesis will argue that:

- The effects of different process characteristics on project performance are poorly understood but such understanding would be beneficial to industry and academia alike.
- Predicting design project performance is a challenge in industry and current trends, such as decreasing development times, are set to result in further problems.
- There is an industrial need for improved tools and/or techniques to support model-building and analysis. Such tools/techniques should focus on the following areas: 1) modelling and representation; 2) variations in scale and connectivity; 3) rework and 4) the product-process link.

A software tool, designed to satisfy this need will be proposed and its development and case-study evaluation, described. The thesis will illustrate that:

- The tool can be used to investigate the impact of different process characteristics and to determine which combinations of structural variations and uncertainties are likely to result in project delays.
- Analyses using the tool lead to improved understanding of design process behaviour, particularly rework and the product-process link.
- Insights from the analysis are beneficial to industry.
- There are number of other potential applications of the tool. For example, it can be used to provide sample data for the development of process visualisation and optimisation tools.

2

Methodology

The design of complex products is a multifaceted process encompassing a wide range of activities. In this context, the use of suitable tools and methods is critical in avoiding schedule slippage and budget overspends, and improving product quality. Likewise, the design of this thesis embodied a broad range of activities and the appropriate use of methodologies was an important factor in delivering high-quality research.

This chapter first describes the overall research methodology for the thesis, explaining how the subsequent chapters are interlinked. Next, the case-study methodology is described and particular emphasis is given to interview and observational-study techniques. Prior to the closing summary, the research methodology for theory and tool development is discussed.

2.1 Overall research methodology

Engineering design research is inherently multifaceted, multi-layered and complex. Success in design research projects often requires multi-disciplinary research, drawing on such diverse fields as psychology, sociology and computer science (Eckert et al., 2003). A design research methodology provides a framework that encourages systematic research and more-repeatable results.

According to Blessing et al. (1995), a methodology should force the researcher to establish the aims and objectives of a given project before the research work is carried out. These aims and objectives can later be used to determine how successful a project has been. However, this can lead to a situation where design researchers focus on solving a problem which does not really exist in industry. Design research, which aims to better understand design and/or produce tools and techniques for industry, encompasses activities addressing four key questions (Cantamessa, 2001).

- How is design performed in industry, in particular within processes we would like to improve?
- How can we understand the cognitive, social and cultural mechanisms that underlie the phenomena we observe?
- What computer tools, pencil-and-paper techniques or design methods might be useful, and how can we develop them?
- How can we introduce these tools or methods into industrial use, and what happens when we do?

Methodologies assist researchers trying to answer these key questions. This work is largely based on the approach taken by Eckert et al. (2003) – which describes a methodological framework to integrate research from different domains – but also draws from earlier work by Blessing et al. (1995).

2.1.1 *Choosing a methodological framework*

Eckert et al. (2003) propose a framework based on the interaction of empirical studies of design behaviour, development of theory, development of tools and procedures and the introduction of these tools and procedures to industry (Fig. 2.1).

- Empirical studies of design behaviour: These can include case studies and a range of analytical approaches, as well as experimental studies of individual design activities.

- Evaluation of empirical studies: This includes assessing the validity of the research results with respect to generalisation and other theories of design behaviour.
- Development of theory: Empirical research should lead to the development of theories and models of design.
- Evaluation of theory: These theories should be assessed both in terms of their philosophical and methodological assumptions and their grounding in more general theoretical frameworks.

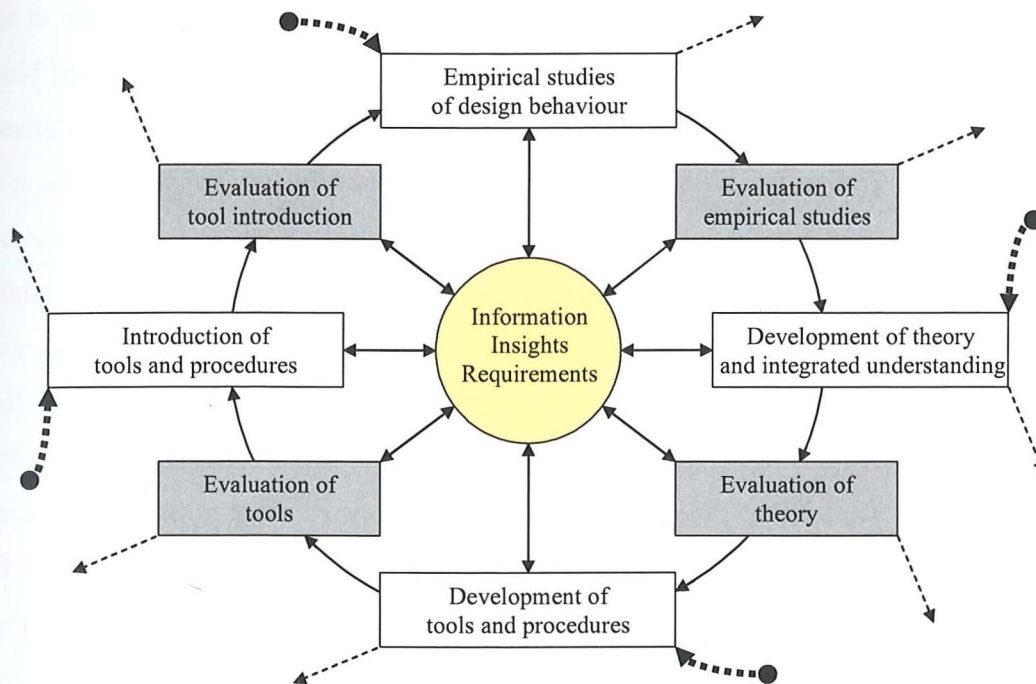


Fig. 2.1 The Design Methodology Spiral (Eckert et al., 2004a). Dotted arrows represent possible entry points for PhD research projects.

- Development of tools and procedures: Computer tools and techniques should be developed to encourage the application of new theories.
- Evaluation of tools: Iterative development and user-testing is required to generate effective tools and procedures.
- Introduction of tools and procedures: Tools and procedures that have been successful during user-evaluation should be observed under real industrial conditions. This is dissemination of research.
- Evaluation of dissemination: Observations from tool introduction and subsequent tool use should be assessed for validity. It is important to verify that tools fit into the general understanding of design practice.

Eckert's framework (described above) is envisaged as an approach for research performed in large research teams and all of the eight steps cannot usually be completed by a single researcher during the course of a PhD thesis. Instead, a complete project usually focuses on a distinct stage or group of stages within of the framework. For example, tool development could build upon earlier research but needs to show awareness of other stages.

An important point is that each stage of the research can provide new insights that lead into any other stage within the framework. In empirical research, for example, one could identify the need for further empirical research into a particular area or it could highlight the need for tool development.

Ideally applied research would form a clockwise cycle, spiralling inwards to converge on a solution space, but in practice research is not so straightforward. Several of the above activities may be performed in parallel, and there is often a need to backtrack if a failure occurs or when new insights are obtained. Nonetheless, these failures can have important consequences such as the identification of new research issues. This is analogous to iteration in design processes, where the solution is refined each time a problem is researched. Also, the research problems evolve with the changing research context both within industry and within the design research community.

While the above approach is aimed at constructing a framework in which multidisciplinary methodological approaches are facilitated, others focus on developing methodologies aimed specifically at design research in the context of PhD projects, where each student works in relative isolation. The design research methodology, DRM, developed by Blessing et al. (1995) is closer to the latter.

This thesis will be written in the Engineering Design Centre as part of a large research group. As such, this work plans to build on and supplement other PhD research performed at Cambridge and will not rigidly follow the DRM approach, aligning itself instead with the design research spiral.

2.1.2 Linking thesis-structure to methodology

The various themes of the spiral methodology act as the backdrop for different chapters of this thesis. An empirical study of design behaviour is the core topic of Chapter 4. Methodological issues concerning the study are described below (section 2.2). The generality and validity of the case study findings are evaluated by comparing them with previous case studies. Chapter 5 considers both practical and

theoretical implications of the case study and the literature review, with respect to supporting design planning. In doing so, it builds on existing theories and models of design while also defining the high-level research requirements. The subsequent chapter concerns the development of a solution concept (to meet these requirements) and its implementation. Evaluation of these theoretical contributions and the software tool is presented in Chapters 7 and 8. The latter chapter assesses the effectiveness of the tool in analysing a real-world model and leads to heuristics for improved planning. Although these chapters (3-8) are arranged sequentially in the thesis, the case study research was carried out largely in parallel to the other chapters; evaluation and iterative refinement of both the tool and its underlying theoretical assumptions taking place in response to feedback from case study participants.

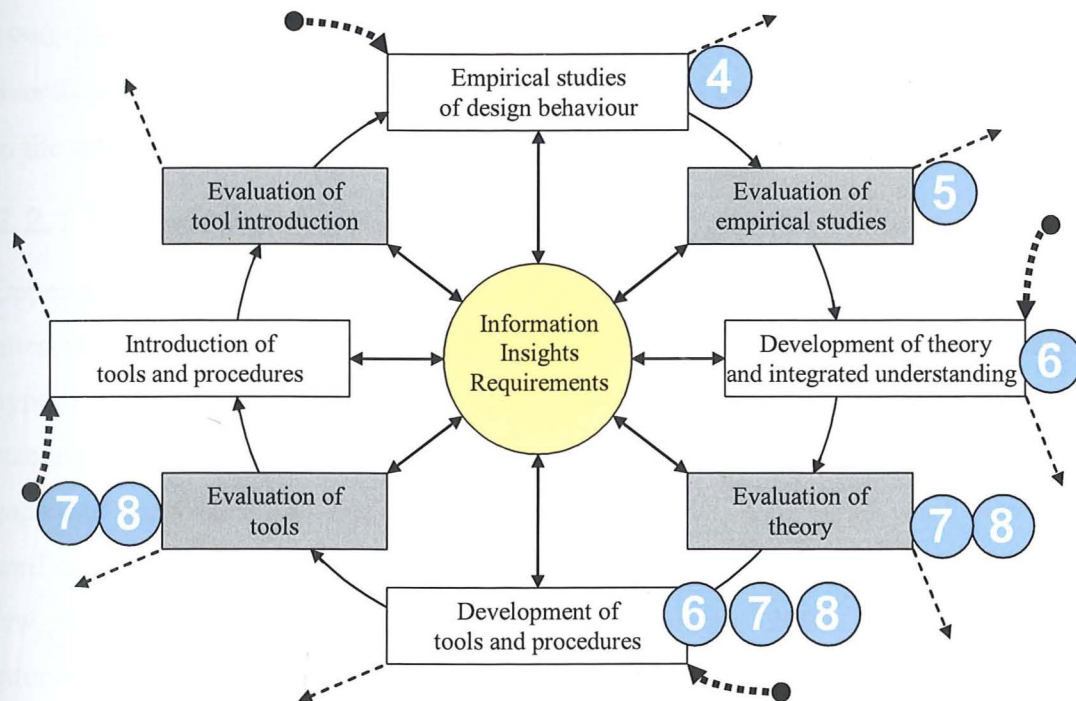


Fig. 2.2 The link between thesis structure and methodology (Chapter numbers in blue circles)

Thorough treatment of all steps in the Design Methodology Spiral is beyond the scope of a single researcher (Eckert et al., 2004a). Nevertheless, this research covers many of these steps while remaining strongly grounded in the findings of other researchers. Because of the broad range of topics considered, the specific contribution to each area is less than would be expected for a single-focus thesis but the total contribution is nonetheless equal or greater.

2.2 Case study

The case study, which lies at the core of this research, was performed with Perkins Engines. It began six months into the PhD project – in April 2003 – and initially played a key role in identifying the research requirements. As the work progressed, over the following 30 months, company feedback was central to the development and validation of the solution proposal.

Data-gathering during the case study was not based solely on a single method; several different methods were used in parallel. This section provides a brief overview of these case study techniques and is followed, in the subsequent section, by details of their application to this research. Each approach has strengths and weaknesses and the strengths of one approach can compensate for the weaknesses of another (Patton, 1990: 244). Triangulation of data obtained from different techniques can improve the overall quality of the case study data (Yin, 1993: 3) and even overcome some bias due to the subjective perspective of the observer.

2.2.1 Interviews

Oppenheim (2000: 65) classifies interviews according to their purpose: exploratory interviews (also termed depth interviews) are used to develop ideas and research hypotheses while standardised interviews are used for surveys, to gather facts and statistics. *“Depth interviewers must ‘listen with the third ear’; they must pick up on gaps and hesitations and explore what lies behind them; as a result, such interviews tend to be unstructured”*. The researcher suggests the subject for discussion but has few specific research questions in mind (Oppenheim, 2000: 67). Standardised interviews assume a semi-structured nature where the researcher introduces the topic then uses specific questions to guide the conversation (Rubin and Rubin, 1995: 5).

Before any interviews take place, it is beneficial, although not always possible, to choose a representative sample of interviewees; otherwise potentially valuable perspectives are likely to be missed and important issues overlooked (Bryman, 2001: 83). During the interview, efforts must be made to remain impartial such that the perspective of the interviewer does not limit the data obtained from the interviewee. The interviewer should strive to ask clear, neutral questions and, when appropriate, use probes and follow-up questions (Patton, 1990: 324). He/she should also be aware of the pitfalls of acquiescence-response-bias – some interviewees may try to bias their

answers to match what they think the interviewer wants to hear (Breakwell et al., 2000: 245). Also, it is important to give appropriate verbal and non-verbal feedback – this helps the researcher to remain in control of the interview and can make the interviewee more comfortable.

After completion of the interview, the researcher commences with transcript analysis. Interviews produce a vast amount of data and, in some cases, much time and effort can be saved by focusing on topics of key interest. For exploratory studies, however, it is also useful to perform inductive analysis in order to identify patterns, themes and categories of analysis that emerge from the data (Patton, 1990: 390). For standardised interviews, statistical analysis of data is often more appropriate (Oppenheim, 2000: 279).

While interviews are a valuable source of research data, they are not without their limitations. Interview data may be distorted by recall error and due to “*personal bias, anger, anxiety, politics and simple lack of awareness*” (Patton, 2000: 245).

Interviews at Perkins

This research was the second case study performed at Perkins by a PhD student from the Cambridge Engineering Design Centre (EDC). The previous collaboration had established a positive working relationship with the company and provided a good introduction to the working practices employed at Perkins. Prior to any company visits, the opportunity arose to listen to recordings of interviews carried out at Perkins and discuss relevant issues with the researchers (PhD student and supervisor) involved. Although these interviews were primarily focused on engineering change management (Jarratt, 2004), they nonetheless provided useful insights into the company and its process planning/management challenges. They were invaluable in establishing contact within Perkins which greatly simplified the process of setting up the interviews for this research (described below).

In total, 46 interviews with 31 designers/managers were carried out during this thesis, each interview lasting between 45 and 90 minutes. The resulting tapes were transcribed and the analysis was fed back to the company for comment. General agreement was reached concerning what constituted the main problems with project planning, although cause and effect relationships of problems and their symptoms were difficult to discern.

Initial interviews

The initial interviews took place in April/May 2003 and covered a broad range of issues including planning, process improvement, information flow, communication, and organisational structure. Engineers and managers were interviewed individually for an average of just over one hour each (Table 2.1). Non-engineering managers represented purchasing, sales, logistics and manufacturing. These interviews were carried out jointly by Dr. Eckert and the author at Perkins' headquarters in Peterborough. The interviews were semi-structured – interviewees were questioned on the same list of topics but were allowed to move the discussion in a particular direction if they deemed this appropriate.

The main aim of these interviews was to better understand communication within the company as a prerequisite to an envisaged relocation. However, they covered a broad range of issues including organisational structure, project management and process improvement initiatives. Thus, the interviews were very beneficial in highlighting suitable opportunities for doctoral research.

Role	Code	Date
Program manager	PrM	15/04/03
Engineering project manager	EPM	15/04/03
Sales manager	SM	22/04/03
Manufacturing manager	MM	22/04/03
Logistics manager	LM1	22/04/03
Purchasing manager	PuM	22/04/03
Large components ¹ team-leader	LCT	01/05/03
Manager of team-leaders	MTL	01/05/03
Electronic systems team-leader	EST	01/05/03
Engineering designer	ED	08/05/03
Product engineering manager	PEM	08/05/03
Rotating components team-leader	RCT	08/05/03
Mechatronics manager	MeM	08/05/03

Table 2.1 Initial interviews

¹ Large components included such items as engine blocks, heads and exhaust manifolds

Problem elaboration interviews

The second set of interviews focused on process improvement for new product introduction with a view towards understanding the design process and identifying associated challenges (Table 2.2; Table 2.3). Two of these interviews were carried out jointly by the author and Dr. Eckert (SM 05/05/04; SSB1 05/05/04), the remainder by the author alone.

Information from these interviews was used to define research problems by identifying planning challenges in Perkins. Interviewees were drawn from engineering management and project planning roles. Again the interviews were semi-structured and different process stakeholders were encouraged to present their individual perspectives rather than stick rigidly to a specific set of pre-defined questions.

Some of the interviewees requested that their interviews not be recorded (Table 2.3). In such cases, notes were taken during the interviews and mind-maps of important comments and issues were constructed immediately after their completion.

Role	Code	Date
Senior manager	SM	05/05/04 07/07/04 27/05/04
Six-sigma blackbelt, ex-process person	SSB1	05/05/04
Process documentation manager	PDM	27/05/04
Finance and accounting manager	FA	14/06/04
Project manager – Perkins Consulting	PMP	14/06/04
Process documentation manager	PDM2	14/06/04
Large components team-leader	LCT	17/05/04
Six-sigma blackbelt	SSB2	17/05/04

Table 2.2 Problem elaboration interviews (audio tape-recordings available)

Role	Code	Date
Manufacturing auditor	MA	03/02/04
Logistics manager	LM2	03/02/04
Electrical engineer	EE	10/02/04
Senior manager	SM	26/03/04 21/04/04 19/05/04 27/05/04
Six-sigma black belt, ex-process person	SSB1	28/04/04
Project office leader	POL	28/04/04
Engine testing planner	ETP	05/05/04
Bill of materials manager	BoM	26/05/04
Risk management manager	RMM	27/05/04
CAE team-leader	CaeT	20/08/04

Table 2.3 Problem elaboration interviews (no recordings available)

Solution evaluation interviews

The final set of audio-recorded interviews took place during the solution evaluation phase of this research (Table 2.4). The range of interviewees was narrower than during earlier interviews, but the content was more focused on the specific contributions of this research in contrast to the initially broad range of issues considered. These interviews concerned the utility of the research findings and the potential for their dissemination within Perkins. The majority of those interviewed at this stage had also been interviewed at an earlier point in the research and were keen to see how their views had been incorporated into the proposed solution. Their appraisal of the research is discussed in Chapter 8.

Role	Code	Date
Senior manager	SM	27/01/05 17/02/05 27/07/05 09/11/05
CAE team-leader	CaeT	11/05/05
Large components team-leader	LCT	11/05/05 19/07/05
First-level manager	FLM 2	11/05/05 09/11/05
Project planner – Perkins Consultancy	PPP	16/06/05

Table 2.4 Solution evaluation interviews

2.2.2 Observational studies

In addition to interviews, this research drew on observational techniques including ethnography and action research to gather data on project planning and management at Perkins. During an ethnographic study, the researcher is consciously aware of the system being observed but does not actively try to change the system. In contrast, action research combines observation with actions designed to effect changes.

“The background of ethnography is wide and it draws on many disciplines. That is why there is no explicit definition of ethnography. In its widest sense, ethnography (etnos = people, race; grafia = the writing, description) is defined as a systematic process, through which models of culture or subculture are observed, described, documented and analysed” (Juntunen, 2001: 43). Ethnography has its origins in anthropology: researchers of foreign cultures lived amongst their study subjects, observing their habits and customs. More recently, ethnography has been used to study different social factions within western societies (Agar, 1996) and within industry (Bucciarelli, 1994; Henderson, 1999).

Ethnographic methods allow the researcher to explore the organisation or group or culture in depth, and with low impact. However, the degree of impact varies with the study: some researchers maintain a distance from their study subjects while others become fully immersed in the culture which they aim to understand. Regardless of the

degree of immersion, however, reflexivity – the fact that the ethnographer is a part of the social world under observation – is almost certain to cause bias (Hammersley and Atkinson, 1995: 17)

One problem with ethnography is that it focuses mainly on observing external behaviours – the observer cannot clearly see what is happening inside people, although external emotions and expressions provide hints. Hence, it is useful to perform interviews in parallel to observational studies, as the latter approach affords research participants the opportunity to make hidden thoughts explicit. Another limitation is that the ethnographer may influence the situation being observed in unknown ways: study participants may behave in some atypical fashion when being observed or the selective perception of the observer may distort the data (Patton, 1990: 244). A more detailed discussion of the merits and limitations of ethnography, along with guidelines for the use of the technique can be found in Silverman (2004) or Agar (1996).

While system distortion due to the observer is considered a limitation of observational study techniques such as ethnography, it nonetheless constitutes an underlying premise of action research: the researcher plays the dual role of actor and observer, alternating between action and critical reflection (Dick and Swepson, 1997). Action research is described by Dick (1999) as a family of research methodologies which pursue action (or change) and research at the same time. It usually involves a cyclic process which alternates between planning, action and evaluation (Waterman et al., 2001). Insights from the evaluation phase of previous action research cycles lead to further actions which aim to improve the system under observation. Theory may be generated and refined as the action research process progresses (Waterman et al., 2001).

Action research methods are most likely to be appropriate when the research starting-point is not obvious and the time available for the study is limited. Action research is suitable in situations where causal explanations are either not possible, or too cumbersome to be useful. When there are many variables, and they interact (often bi-directionally) in complex ways, causal explanations are themselves likely to be very complex (Dick and Swepson, 1997). A limitation of action research is that it often tends to yield qualitative rather than quantitative results and these results are not

highly-generalisable. On the positive side, it can lead to specific, local understanding which is overlooked by other techniques.

This research involved both ethnography and action research. The initial phase was ethnographic in so far as it observed planning practice with the goal of increasing academic understanding of how industrial projects are planned. The final phase was more closely aligned to action research because it was designed to effect planning changes.

Observational study at Perkins

Four weeks were spent on site at the company, observing different employees as they planned and executed the design of a diesel engine. As with the interviews, the observational study was composed of an initial study concerning problem identification and the in-depth analysis of project planning.

During February 2004, two days were spent shadowing a logistics manager who dealt with issues from supplier sourcing, supplier removal, deletion of outdated parts, lead-time reduction and dual sourcing to logistics team meetings and project reviews. Subsequently, two days were spent with a programme manager, who endeavoured to prevent problems from impacting the manufacturing line and to deal with new problems as they arose.

Between May and July 2004, a total of three weeks were spent observing the head of the project planning office and his staff. The project office team, which consisted of six employees, was the central planning group in Perkins responsible for integrating plans from individual team-leaders. During this time, opportunities arose to talk with six different project plan administrators concerning their views on planning within the company. These conversations are not listed in the above section on interviews because of their informal and brief nature. Observations from this part of the study provided insights into the way in which the company created and maintained plans, who was involved in the planning process and what attitudes different stakeholders had to plans. In addition to observing employees performing routine tasks, the author also attended several company meetings during the observational study. These meetings covered a wide variety of topics ranging from supply chain logistics issues and manufacturing delays to project plan updates and provided insights into the company culture.

2.2.3 Documentation

Bryman (2001: 395) notes that documents constitute a heterogeneous set of sources of data, ranging from personal documents, software files, and graphic representations, to official documents from both state and private sources. However, the volume of documents can be considerable and thorough analysis of all relevant documents may be beyond the scope of a particular research project. Also, it is important to consider the quality of information contained in documents as they are written with a particular purpose in mind and many omit information which is important for alternative applications. Criteria for evaluating information based on documentation are authenticity, credibility, representativeness and meaning (Bryman, 2001: 396). Failure to account for potential bias and spin-doctoring can lead to incorrect inferences in the absence of sufficient collaborative evidence.

Nonetheless, document analysis should not be underestimated as a potential source of valuable information which can complement other case study data: *"it provides a behind-the-scenes look at the programme that may not be directly observable...the interviewer might not ask appropriate questions without the leads provided in the documents"* (Patton, 1990: 233). The utility of documentation, as a source of data on Perkins, is discussed below.

Perkins Documentation

Meeting minutes, project plans, process documentation and project review literature all provided useful information on how Perkins plans its engine development programmes. Some of these documents overlap each other, some are slightly contradictory but they nonetheless contain valuable data which can be compared against that from other sources. While planners, managers and designers have a good overview of projects and the tasks which comprise them, documents such as project plans were free from some of the biases which arise from interviews and observational studies.

2.2.4 Summer student collaboration

During the summer of 2005, Perkins hired an intern from the University of Cambridge to evaluate the practical challenges of implementing recommendations from this research within the company. During her nine-week placement, the student typically spent three days each week in Cambridge and two days on site with Perkins.

The arrangement avoided a lot of problems with security clearance, copyright and intellectual property rights. Because the student was an employee of Perkins, she had access to information which was normally considered sensitive. Also, she had access to the company's computer network and could easily check the availability of different company employees and thus arrange meetings more efficiently.

Information from the case study was used to define the research requirements and to guide the development of tools and techniques to support industry. The methodology for this research phase is described below.

2.3 Theory and tool development

As with design processes in general, this research was envisaged as an iterative process; following an initial period focused on defining the problem empirical studies, theory development and tool building activities were expected to take place largely in parallel. Insights from each activity drove new research in the others (Fig. 2.3). Further, all of these activities built on existing research.

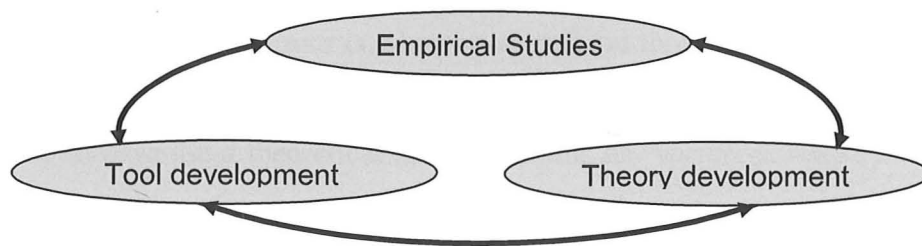


Fig. 2.3 The iterative process of empirical studies, theory and tool development

2.3.1 Building on existing research

The Design Methodology Spiral (Fig. 2.1) is aimed primarily at large research groups, which are in a position to pursue long-term research agendas and tackle fundamental questions. This research aimed to build on and interlink with other work undertaken in the EDC (Section 1.2.2). Despite the temptation to move on to completely new areas, the merits of group interaction rather than purely individual research are considerable. Previous research within the EDC has also focused on process modelling and analysis. Research into Signposting has been an ongoing theme within the group for almost 10 years and four previous PhD students had worked on different aspects of Signposting prior to the commencement of this thesis (see section 3.4.3). The industrial studies performed as part of these PhDs together demonstrated the generality of process

improvement challenges in industry and provided useful insights into the nature of these problems. The research reported in this thesis aims to build on these findings.

2.3.2 Theory development

Theory-development is an important part of creating new knowledge. However, before discussing how the theory for this work was developed, it is useful to provide a definition. Morrison (2003) defines theory as follows: *"Theory has a definite structure. It comprises variables, relationships between these variables, and a carefully explained logic underlying these relationships. Theory is also not an art of description; it is the art of prediction and explanation of relationships in the empirical world"* (Morrison, 2003). For the purposes of this research, the above definition was considered appropriate with the following qualification: it fails to account for the fact that design is a social process (Minneman, 1991) and hence that theory development in design often diverges from disciplines such as physics where key variables are of a more concrete nature.

Dubin (1969) points out that it is possible to distinguish between an empirical system and a theoretical one: the former is what we apprehend through human senses, while the latter is the mental representation that we construct to model the empirical system. He goes on to describe a theoretical model in terms of *"variables whose interaction constitute the matter of attention"* along with a specification of how these variables interact. Building on this work, Whetten (1989) defines theory in terms of 1) what, 2) how, 3) why and 4) who-where-when. 'What' concerns the variables, constructs and concepts that are used to define a theory. Choosing these theory building blocks involves a trade-off between comprehensiveness and parsimony. 'How' concerns the relationship between different elements and typically introduces causality. 'Why' concerns the deduction processes underlying the theory and should be falsifiable – 'what' and 'how' describe, only why explains. 'Who-where-when' defines the generalisability of the theory. *"During theory development, logic replaces data as the basis for evaluation"* (Whetten, 1989). Bacharach (1989) developed a visual representation for the components of a theory (Fig. 2.4) which shows how variables and constructs differ in terms of their abstractness and how they are respectively linked through hypotheses and propositions. This research aimed to contribute to design theory by increasing process understanding and by providing insights into planning in industry.

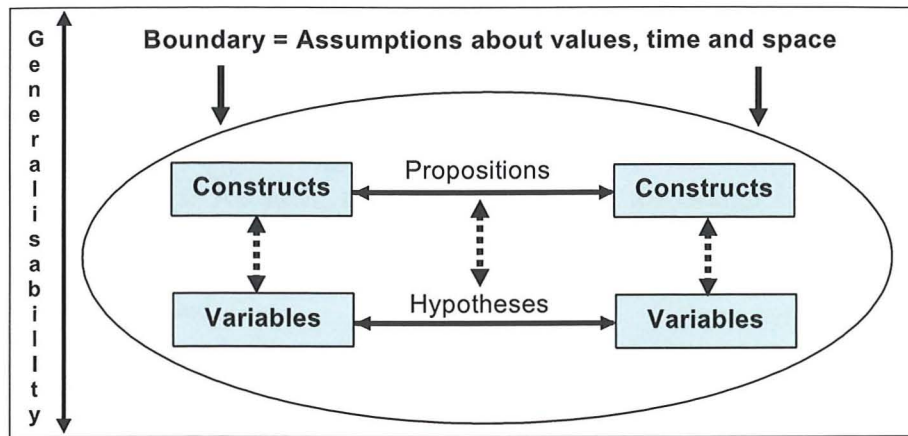


Fig. 2.4 Components of a theory (Bacharach, 1989)

2.3.3 Development of tools

Developing a Signposting software tool was considered an important part of this research, especially when considered in terms of the overall EDC's research strategy. As noted in section 2.3.1, the development of the Signposting framework has been an ongoing theme in the EDC for several years. Each thesis has produced a new software tool. It is intended that all of these software tools will eventually be integrated, establishing a single toolkit for continued collaboration with industry. An EDC software programmer has worked concurrently on these and other EDC projects with the dual goals of co-ordinating the integration effort and supporting the implementation of three different Signposting tool-variants. Another PhD student also contributed significantly to the implementation of the Signposting software tool described in this thesis before eventually deciding to focus on visualisation of engineering change propagation.

Although previous research had resulted in different incarnations of Signposting tools (in Fortran, Lisp and C++), it was decided to use Java as the programming language for this work because of the following reasons: 1) Java is platform independent; 2) useful tools and documentation for Java are plentiful; 3) it prevents common programming errors such as misuse of pointers and memory management, thus leading to more reliable code; 4) it is a popular, modern Object-Oriented language, a factor which increases the chances that it will appeal to future researchers; 5) several members of the EDC's Design Process Improvement group are familiar with Java and prefer it to other programming languages. Although some concerns have been raised about the speed of Java, its benefits nonetheless outweighed its disadvantages.

While several software design methodologies exist (see Sorensen, 1995, for a comparison of waterfall, incremental and spiral methodologies), many of these are aimed at companies developing commercial software in large teams, subject to well-defined requirements. As such, their appropriateness for design research is limited both by the scale of such projects and the fact that the requirements were evolving as the research progressed.

To overcome the limitation of generic methodologies, a design-specific software-development-methodology has been proposed by Bracewell et al. (2001). This provided useful guidance on the appropriate objectives for tools constructed in a research context. Also, Bracewell's methodology considers tool development in the context of other research activities, such as data gathering (as discussed in section 2.2), and evaluation in context, the topic of the subsequent section.

2.3.4 Application in context

The research tool and the resulting heuristics for improved planning were evaluated through empirical study. Development and evaluation of professional software and its introduction into industry was considered beyond the scope of this work. Nonetheless, it was important to demonstrate proof-of-concept for the tool in order to engage continued support for future research: such support can eventually lead to more robust software implementation if the underlying methodological and philosophical assumptions are valid.

This research aimed to contribute to academic understanding of design processes by developing the Signposting modelling and simulation-analysis tool and by applying it to explore design processes. In addition to this academic contribution, this research aimed to provide practical benefits to industry. Sometimes academic and industrial goals can be in slight conflict: industry striving for short/medium-term, focused, easily-applicable recommendations for process improvement, while academia is willing to explore more theoretical, long-term issues with the same ultimate objective. In order to perform the industrial evaluation, both the tool and the resulting heuristics were presented back to Perkins and different employees were interviewed on the merits and limitations of both research contributions. Some of the evaluation-work was performed by the summer student – the fact that Perkins was willing to pay for her summer placement constitutes evidence for the practical value of the research.

2.4 Summary

This chapter presented the methodological foundations for the thesis. It first showed how remaining thesis chapters are aligned with the design methodology spiral. This was followed by a review of data-gathering techniques for the case study; their application to this research was also described. A discussion of methodological issues, relating to theory and tool development, concluded the chapter.

3

Research into planning and modelling complex design projects

The design failures described in the introductory chapter (the Vasa tragedy, the Ariane 5 explosion, the Ford/Firestone saga, the Tacoma Narrows bridge-collapse and the cost and schedule overruns of the MoD) together illustrate some of the key challenges involved in modelling, analysing and planning complex projects. This chapter reviews several different research strands which are relevant in addressing these challenges.

The chapter contends that engineering design processes exhibit many of the characteristics of complex systems; it opens with a review of literature on complexity and then focuses on complexity in engineering design. In order to better understand which aspects of complexity research are applicable to the analysis of design processes, the characteristics of these processes must be considered. A review of generic design process models was undertaken to shed light on this issue. Despite the limitations of such models, the review nonetheless shows why design planning is a complex-system behaviour-prediction problem.

Planning is not only problematic in design but also in other areas such as logistics, robot motion planning and manufacturing scheduling. Research into Operations Research (OR), simulation analysis and Artificial Intelligence (AI) planning has been undertaken with the goal of addressing these challenges. The applicability of this work to the field of design is discussed. A common theme among different approaches to planning and analysis is the use of models, the properties of the model greatly influencing the insights which can be obtained. The penultimate section of this chapter compares and contrasts different models used for design planning and selects the modelling approach for this work. The chapter concludes by identifying opportunities to support design planning.

During this research, several other topics were also considered including literature on communication in design, academic work on expertise and research into engineering change. These topics are not discussed here as they were considered peripheral to the core thesis topic.

3.1 Complexity

The section examines literature on complex systems in order to better understand the impact of such issues as uncertainty and/or connectivity on plannability. In so doing, it aims to present a more theoretical perspective on the nature of challenges in design planning. Despite the existence of a considerable body of literature, a unified view on complexity is elusive – different authors take alternative perspectives within the context of different disciplines. Nonetheless, there are a number of recurring themes in the complexity literature which provide insight into the nature of complex systems and the associated challenges of modelling and behaviour prediction (Fig. 3.1). These themes are elaborated below.

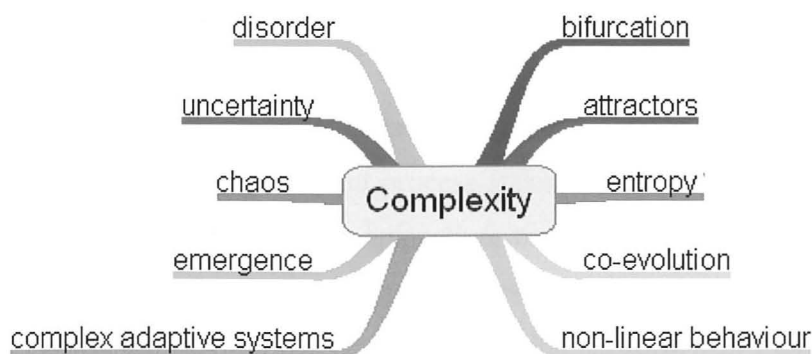


Fig. 3.1 Different themes in the complexity literature

Planning is not only problematic in design but also in other areas such as logistics, robot motion planning and manufacturing scheduling. Research into Operations Research (OR), simulation analysis and Artificial Intelligence (AI) planning has been undertaken with the goal of addressing these challenges. The applicability of this work to the field of design is discussed. A common theme among different approaches to planning and analysis is the use of models, the properties of the model greatly influencing the insights which can be obtained. The penultimate section of this chapter compares and contrasts different models used for design planning and selects the modelling approach for this work. The chapter concludes by identifying opportunities to support design planning.

During this research, several other topics were also considered including literature on communication in design, academic work on expertise and research into engineering change. These topics are not discussed here as they were considered peripheral to the core thesis topic.

3.1 Complexity

The section examines literature on complex systems in order to better understand the impact of such issues as uncertainty and/or connectivity on plannability. In so doing, it aims to present a more theoretical perspective on the nature of challenges in design planning. Despite the existence of a considerable body of literature, a unified view on complexity is elusive – different authors take alternative perspectives within the context of different disciplines. Nonetheless, there are a number of recurring themes in the complexity literature which provide insight into the nature of complex systems and the associated challenges of modelling and behaviour prediction (Fig. 3.1). These themes are elaborated below.

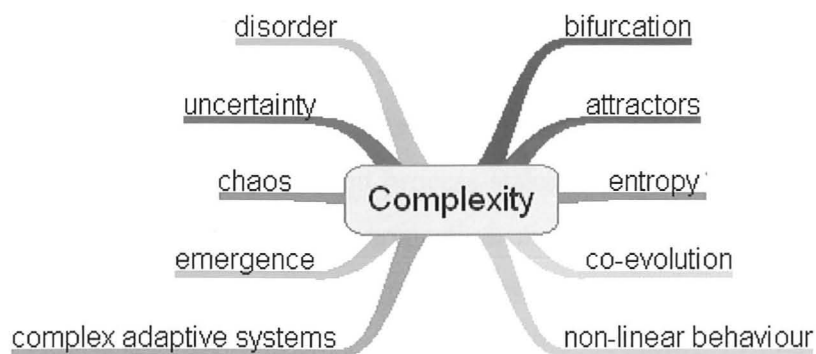


Fig. 3.1 Different themes in the complexity literature

The word complexity is derived from the Latin *complexus*, past participle of *complector*, to entwine. Dictionary synonyms for complex include complicated, interconnected, intricate, intertwined, involved and tangled. These adjectives mean having parts so interconnected as to make the whole perplexing (Simpson and Weiner, 1989).

Despite the value of these definitions as a starting point and the existence of numerous well-founded academic theories, the term complexity eludes strict definition and it can be argued that none of these definitions entirely captures the meaning of complexity. According to Seth Lloyd, as referenced by Nam Suh (2001), there are about three dozen ways in which scientists use the word complexity. Complex systems, as a research field has relevance to many research domains including physics (Gell-Mann, 1995; Prigogine, 1997), computer science (Wilf, 2002), systems theory (Mittleton-Kelly, 2000), social sciences (Byrne, 1998), psychology (Streufert and Swezey, 1986) and engineering design (Earl et al., 2004).

Issues also arise concerning the objectivity of complexity. Some authors (e.g. Streufert and Swezey, 1986) argue that complexity depends on the subjective characteristics of the individual and that the same system may be perceived at differing levels of complexity by different individuals. Conversely, other definitions of complexity (Gell-Mann, 1995; Feldman and Crutchfield, 1998) are objective and concern the level of difficulty in solving mathematically posed problems as measured by the time, number of steps or arithmetic operations, or memory space required. These types of complexity are called time complexity, computational complexity and space complexity, respectively.

Johnson (1995) considers complexity as the combination of an objectively complex backcloth of highly interlinked parts upon which a subjective complexity, reflecting the actions of system-users, is superimposed. This approach is appropriate when considering large-scale design projects: such projects are objectively complex as they comprise large numbers of tasks, resources and uncertainties which are all intertwined. At the same time, different process-stakeholders interact differently with the process depending on their role, perspective, experience and expertise and may also be affected by the description of the process. Actions can take place on the backcloth, either directly or indirectly via descriptions, as shown in Figure 3.2 – the former category of changes tend to be more radical (Eckert et al., 2005).

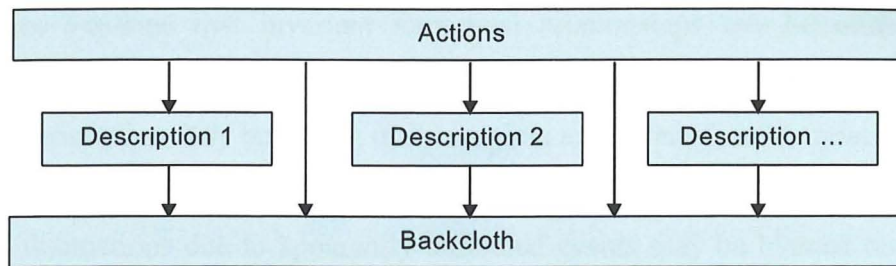


Fig. 3.2 A view of complexity as actions on a backcloth (Eckert et al., 2005)

3.1.1 Non-linear behaviour and bifurcation points

Waldrop (1993) defines complexity as the “domain between linearly determined order and indeterminate chaos” (Fig. 3.3). Chaos is defined as “behaviour of a system which is governed by deterministic laws but is so unpredictable as to appear random, owing to its extreme sensitivity to changes in parameters or its dependence on a large number of independent variables; a state characterized by such behaviour” (Simpson and Weiner, 1989). A linear relation exists between two parameters if a change in one parameter is directly proportional to a change in another for all values of both parameters. For example, Ohm's law states that the potential difference across an ideal conductor is proportional to the current flowing through it. The constant of proportionality is the resistance.

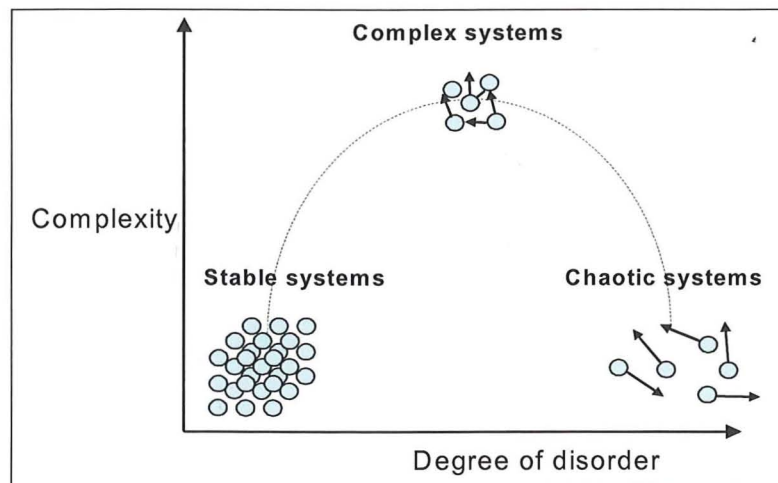


Figure 3.3 The domain between order and chaos (Evalife, 2005)

Streufert and Swezey (1986) add several notes of caution concerning linear models of connected parameters (many of which are also valid for other mathematical models), notwithstanding that such models have useful applications. They point out that it is

sometimes assumed that invariant numerical relationships can be established to represent observable events and that these relationships will hold over time. They note that this assumption may be wrong or incomplete and, even if such a relationship can be established, it may be impossible to accumulate the necessary input information because fluctuations due to apparently unrelated events may be blamed on errors in measurement.

Prigogine (1997) reaffirms these views concerning predictions based on models that assume linear behaviour. He uses the example of a thermodynamic system to illustrate the concept of instability. Only a single solution exists for the thermodynamic system which corresponds to thermodynamic equilibrium (a state in which the thermodynamic variables are stationary) and maximum entropy (entropy is a measure of disorder or randomness in a closed system) – this solution is known as a thermodynamic branch. The branch generally becomes unstable at some critical distance from equilibrium. This point is the bifurcation point (Prigogine, 1997) as shown in Figure 3.4. Far from equilibrium, behaviour in thermodynamic systems is difficult to determine due to the existence of bifurcation points (Prigogine, 1997). At crucial bifurcation points, a non-linear system seems to have two possible trajectories into which it can move. It chooses between them based on very small differences in controlling parameters at the point of change, e.g. weather, butterfly effect.

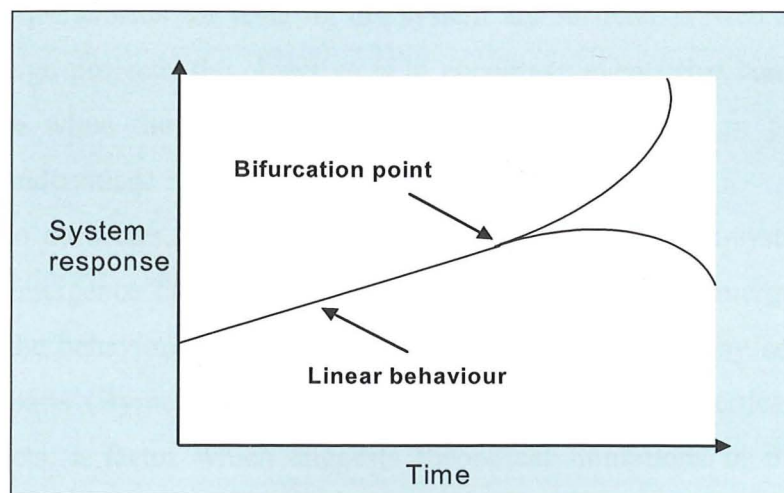


Fig. 3.4 Bifurcation point

The notion of bifurcation points finds parallels in design processes where small changes to the system – e.g. decisions to outsource components, delays to tasks, decisions to use platform components – result in a different route through the process. In complicated processes, tasks can have multiple outcomes and the process can split

in several different directions at a given point – they display “multi-” rather than bifurcation. Further, design processes may embody multiple points of divergence – hence predicting process behaviour is extremely difficult.

3.1.2 *Attractors and emergence*

Despite the existence of bifurcation points, complex systems are not chaotic. Attractors contribute somewhat towards understanding why complex systems remain partially deterministic. An attractor is a set of properties toward which a system tends to evolve, regardless of the starting conditions of the system (Byrne, 1998). For example, the world’s climate can be attracted towards an ice age or towards the current, milder climatic attractor, but is unlikely to stabilise at in-between points.

The author is not aware of any literature concerning attractors in design processes. Nonetheless, it makes intuitive sense that the designers may be attracted to existing solutions from previous designs, even when other equally valid solutions exist, and that this approach will be reflected on the design process. Also, plans, once created, can act as attractors: humans often make concerted efforts to recover delays and stick to plans, even when unexpected events occur.

Although it is not possible to predict what will happen for complex systems, one can act proactively to encourage certain events and discourage others, assuming that the appropriate mechanisms for leveraging the system are sufficiently well known. When planning design projects, the objective is to encourage events that constitute project success even when the complex system in question, the design project, is not completely understood.

In contrast to attractors, which can simplify the task of complex-system behaviour prediction, emergence can thwart the best efforts of planners. Emergent properties arise when the behaviour of the whole cannot be accounted for by considering the sum of its parts (Byrne, 1998). In many cases, emergent properties arise due to holistic effects, a factor which suggests theoretical limitations in the prediction-capability of models: unless the models completely capture the system which is being examined, they are unlikely to correctly predict emergent behaviour. Even for a single design project, however, capturing all relevant influences is not possible, both due to the time it would take to record the elements that can be identified, and because of the impossibility of identifying the full set of significant factors (which include technical, social and political issues).

3.1.3 Complex adaptive systems

Another variation on complexity is described by Mitleton-Kelly (2000) who considers Complex Adaptive Systems (CAS) as a model for business process re-engineering. Complex adaptive systems are dynamic systems able to adapt and change within, or as part of, a changing environment. Each system is closely linked with all other related systems making up an ecosystem. Thus the environment evolves as a whole and each change can be seen in terms of co-evolution with all other related systems. A more extensive account of co-evolution is given by Byrne (1998). He argues that complex adaptive systems, operating on the edge of chaos, are more likely than stable systems to achieve performance improvements because they explore a larger search space. However, the behaviour of complex adaptive systems is extremely difficult to predict; hence the direct applicability of the approach to design planning is limited.

3.1.4 Information, entropy and complex system resetting

Suh (2001) takes an information entropy (see section 3.1.1) view on complexity, defining it as the measure of information uncertainty. He also introduces the concept of time-dependent combinatorial-complexity and time-dependent periodic-complexity. The former is a function of all decisions made over the past history of the system – i.e. cumulative complexity – while the latter assumes that the complex system resets itself to the initial conditions after a given period.

Hence, time-dependent periodic-complexity is not sensitive to uncertainties which arise during previous periods and such problems are easier to solve. Examples exist in nature – organisms die and new ones are born; the atomic structure is periodic. Aeroplane and rail schedules are also reset at the end of each day, removing the uncertainties created over the previous day. Conversely, time-dependent combinatorial-complexity results in large problems which are difficult to model and predict. In addition, combinations of both types of complexity are possible; daily weather reflects time dependent combinatorial complexity while yearly weather repeats periodically. From an abstract viewpoint, the notion of periodic resetting shares some similarity with attractors as both approaches concern system behaviour which resets itself and repeats over time. While Suh focuses on product complexity, his ideas are nonetheless useful for process complexity.

3.1.5 Relating different views on complexity

Eckert et al. (2005) use the notion of actions taking place on a backcloth to explain the relationship between different views of complexity. Broadly speaking, the more dynamic characteristics of change – co-evolution, adaptation – reside on the actions layer. Chaos, in contrast is more concerned with the structure of the backcloth and the resulting behaviour in terms of attractors and emergence (Fig. 3.5). Likewise the concept of information entropy is concerned with connectivity within the backcloth.

At the same time, however, it is important to note that analysis of complex systems usually takes place on models of systems and not on the systems themselves. As such, the way in which the system is modelled has a major influence on the insights that can be obtained; different models and descriptions capturing different aspects of the system. Alternative descriptions of complex systems, which capture different aspects of complexity, can be used together to perform system analysis. As more information about the system is obtained, new descriptions can be created and existing descriptions modified. Problems are likely to arise, however, when descriptions have insufficient scope to describe the system backcloth or when modelling errors lead to inconsistent descriptions (Eckert et al., 2005).

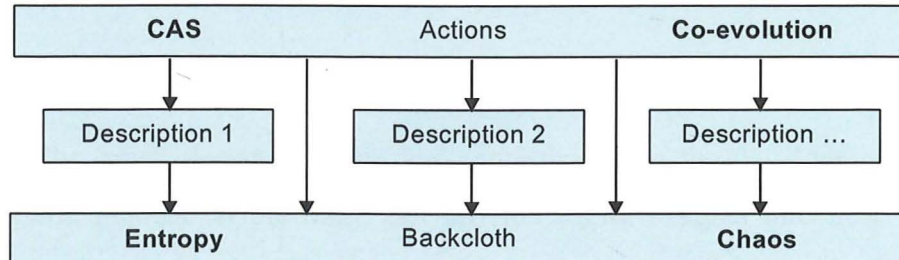


Fig. 3.5 Relating different views on complexity (Eckert et al., 2005)

The next section considers the complexity of large-scale design processes. It argues that such processes exhibit many characteristics of complex systems and considers the resulting implications for their planning and management.

3.2 The complexity of design processes

Streufert and Swezey (1986) point out that predicting complex system behaviour under unstable conditions involves considering many possible, reasonable and meaningful interpretations of events and their likely consequences. *“With uncertainties given, with unknowns and unknowables, with insufficient information*

about interrelationships among a series of uncertain events” predicting behaviour in complex systems is difficult. (Streufert and Swezey, 1986). This echoes work by Earl et al. (2004) who state that predicting design process behaviour is challenging because uncertainty is omnipresent.

Hence, planning – especially in engineering design – is a difficult problem because it requires the prediction of project behaviour under conditions which cannot be determined *a-priori* by humans whose actions are not always foreseeable. Further, many existing approaches to project planning model only part of the system – for example, ignoring resources, constraints, dependencies – and hence fail to capture project complexity. Even when models do capture all of the necessary parameters, dealing with uncertainties and unknowns still poses a challenge. Thus the application of complexity research to design process planning is appropriate.

This section reviews the relevant literature with the goal of identifying characteristics of design processes that affect predictability. It begins by examining generic models, which nonetheless capture important characteristics such as iteration, and concludes by comparing design process characteristics with those of the complex systems reported above. The implications of complexity research on planning design processes are discussed.

3.2.1 Generic models of design processes

Traditionally, much design research has concentrated on the development of high-level generic models. While these can provide useful insights into how processes work at an abstract level and perhaps yield some practical guidance in the form of checklists for design targets, their generic nature limits their value in understanding design processes and hence they receive only a brief description in this thesis. More thorough reviews of design process models are provided by Wynn and Clarkson (2004) and Browning and Ramasesh (2005).

Numerous generic design process models have been proposed. Pahl and Beitz (1996) (Fig. 3.6) and Dym (1994) presented a staged model of the design process in terms of idea generation, conceptual design, embodiment and detailed design. Cross (1989) gives an overview of this model which provides useful guidance for the development of milestones, but does not address specific design activities or product properties. An indication of necessary task sequences can be obtained from activity models (Evans,

1959, Shigley and Mischke, 1989; Blessing, 1994), but these models do not describe a specific process. Bichlmaier (2000) added the concept of generic building blocks to represent typical activities of the design and manufacturing processes. These models emphasise the links between design, manufacturing and assembly, but are of limited use due to their high level of abstraction. Another criticism is that these models do not capture all of the designer's activities: Austin et al. (2001) claim that the model from Pahl and Beitz only made up 47% of design team work (although their study was based on design in the construction industry while Pahl and Beitz concerns engineering design). The remaining time was spent in activities such as project management and organisation.

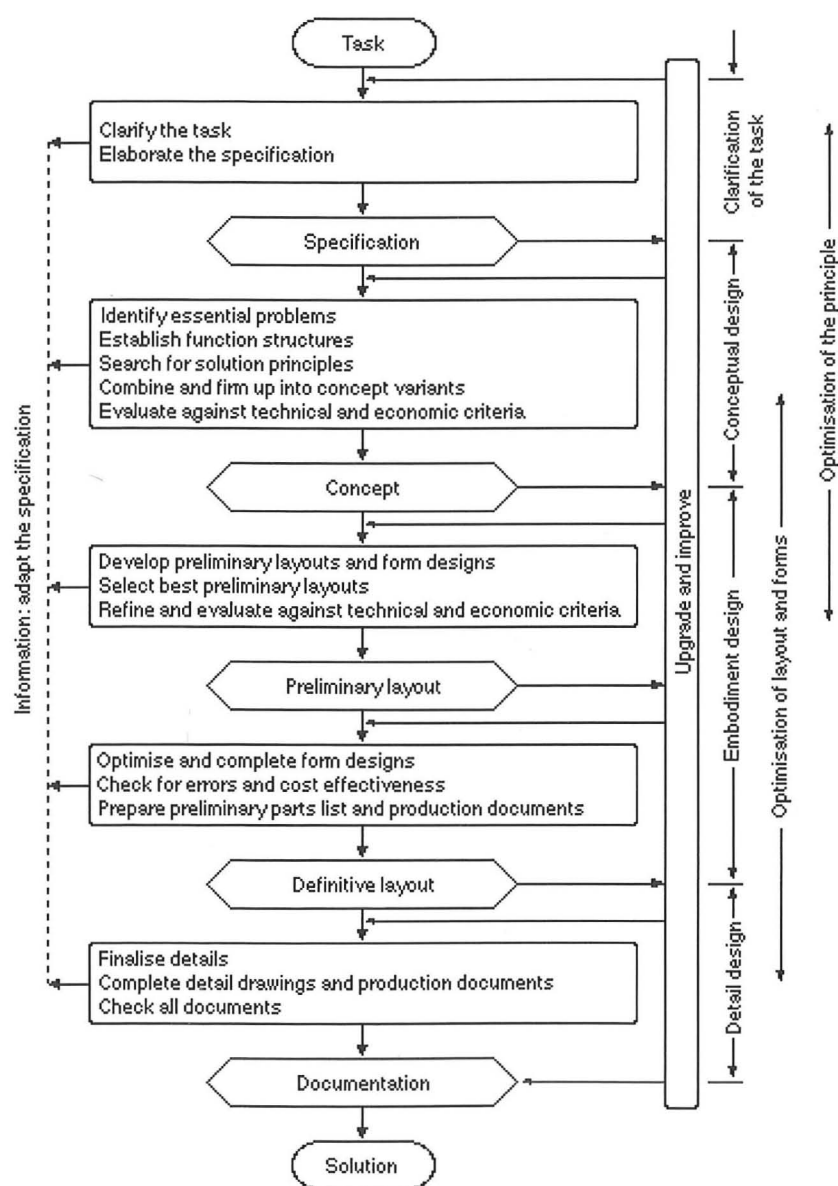


Fig. 3.6 The Pahl and Beitz design process model (Pahl and Beitz, 1996)

Generic models of design processes differ in origin, objective, content and representation approach. Their variety is perhaps a testimony to the challenge of modelling and representing design processes. Despite their diversity, they frequently share common core-characteristics. Several different models use feedback arrows to show dependencies and iteration between different design phases. The notion of converging on a solution – starting with vague ideas and finishing up with a physical product – is also common. There is much uncertainty associated with the design process: even the product specification is being adapted as the design progresses (Fig. 3.7) – a factor which can introduce uncertainty to all other aspects of the design process. Also, many of these models are product orientated in so far as they consider the design process from the perspective of the product which is being designed, rather than design as a social process concerning negotiation between different participants (Minneman, 1991).

In their aim to be generic, such models fail to capture issues which are industry-specific, product-specific or team-specific. It makes intuitive sense that the design process for an aeroplane will create different challenges from those associated with designing a bicycle, but issues such as scale and product complexity are not reflected in generic models.

3.2.2 Characteristics of design processes which lead to planning challenges

Based on the above process models and other research into design processes, it can be seen that core characteristics of design processes include uncertainty, rework and product-process interdependencies. These characteristics lead to challenges in modelling, planning and analysing design processes.

Iteration and rework

The design process models described above (Section 3.2.1) frequently use feedback arrows between different design stages, to symbolise iteration and rework. Iteration may take place between interdependent tasks within a particular stage of the process (Clarkson and Eckert, 2004). However, the generic models do not contain information on task timing or duration and fail to provide guidance on how different feedback loops influence project plannability. Further, they do not distinguish between different

types of iteration, such as planned iteration to converge on a solution, reoccurring tasks and rework due to task failure (Wynn et al., 2005).

Iteration to converge on a solution Iteration can take place in order to converge on a solution (Evans, 1959; Fig. 3.7). When several interdependent components are being designed, information from the design of each component is required to finalise the design of the others. For example, the engine block and the head within a diesel engine exhibit a high level of dependency, such that iteration is common during the design of both components.

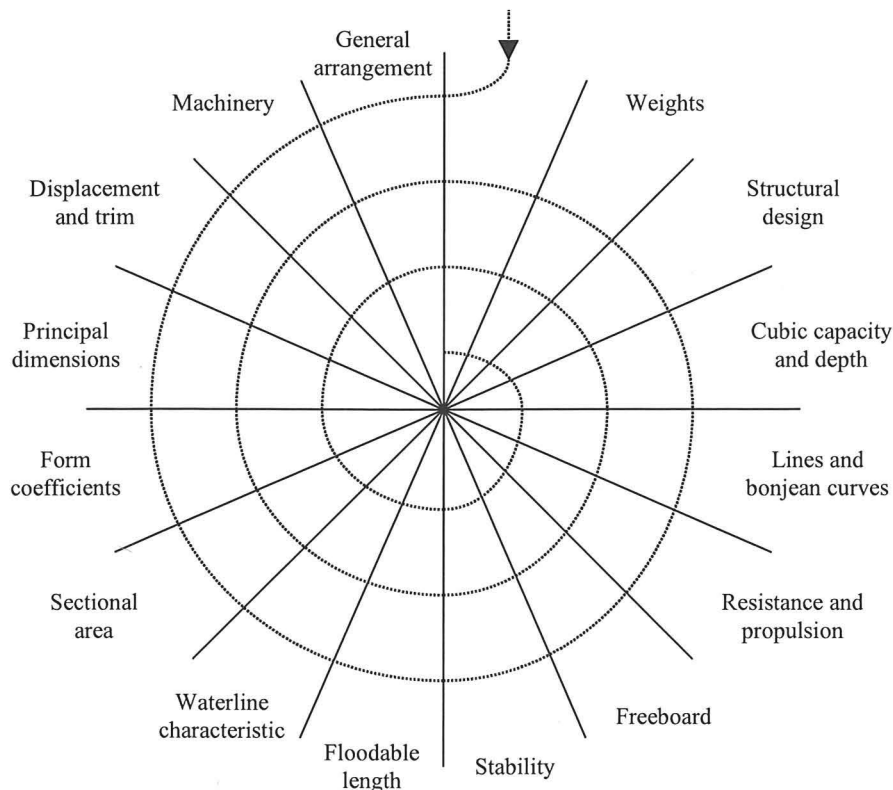


Fig. 3.7 The ship-design spiral (Adapted from Evans, 1959)

Reoccurring tasks Some tasks may be repeated throughout the process. For example, review meetings may take place at the end of each month. However, the design context is likely to vary significantly as the project progresses, and actual work content of reoccurring tasks may vary significantly as a result. In contrast, several very similar tests may be required in order to verify that a product meets its objectives. Some testing tasks are repeated at different points in the design process.

Rework due to undesired task outcome For the purposes of this thesis, rework is defined as the type of iteration which takes place due to task failure. If tasks are not

completed to sufficient levels of quality – a scenario which may arise due to inexperience or time pressure – they may require rework later in the process. Often, such rework is highlighted by testing tasks; when tests fail to verify the design, the root causes for the undesired test-outcome can be traced back to upstream design tasks, some of which must be reworked. Planning for rework is not straightforward. Forecasting which tasks will require rework, predicting how long the rework will take and foretelling the order in which tasks will be executed are all difficult.

The Design Structure Matrix community (www.dsmweb.org; Section 3.4.1) use matrices to capture and represent process dependencies. Useful characteristics of such matrices are their capability to make iteration explicit and to highlight opportunities to reduce iteration by improving the task order. Yassine et al. (2001) use DSMs to model rework in the automotive industry and to assess project sensitivity to errors in rework likelihood estimates. Cho and Eppinger (2005) discuss how reworked tasks may be performed in sequence, in parallel, or with varying degrees of overlap depending on the extent of information dependency between tasks. They note that such models of rework, while useful, are not without limitations such as poor scalability and oversimplifying assumptions.

Cooper (1993) presents the rework cycle as a mechanism to explain major delays, especially in large scale projects (Fig. 3.8). Based on extensive industrial experience as a consultant, he argues that undiscovered rework, creates a falsely optimistic picture of progress and that early discovery of such rework is critical to on-time project completion. He also points out that late augmentation of resources often does little to improve the situation and that, in some cases, the approach can even exacerbate the problem.

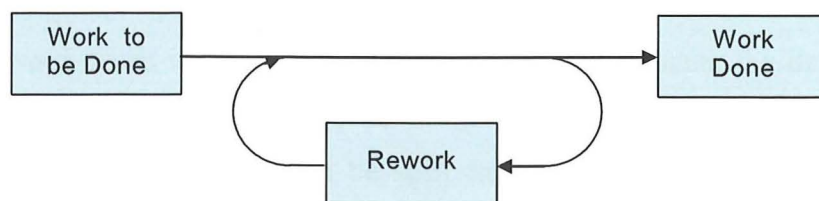


Fig. 3.8 The rework cycle (Cooper, 1993)

While different authors have highlighted the importance of rework in engineering design, extensive analysis to determine the significance of different rework mechanisms, especially in combination, has not been performed.

Uncertainty

Design projects contain numerous sources of uncertainty (Browning, 1998). Task durations, task order, task outcome, rework likelihood, resource availability, sales volumes, supplier performance, customer requirements, the likelihood and impact of engineering changes (Nichols, 1990), and even such diverse issues as governmental decisions on taxation policy are often uncertain during the project planning phase.

Broadly speaking, uncertainties can be classified as known unknowns and unknown unknowns. For example, companies know that some tasks are likely to be delayed but they do not always know why or by how much. In contrast, the war in Iraq is an example of an unknown unknown from the viewpoint of diesel engine sales – it was difficult to predict whether the war would take place and the resulting impact on diesel engine sales was also difficult to predict. While it is possible to perform some kind of risk analysis for known unknowns, dealing with unknown unknowns is particularly challenging. Another important issue, when dealing with uncertainty, is differentiating uncertainty in the model from uncertainty in the process, as discussed in section 5.1.

Although the existence of uncertainty is widely acknowledged by both industry and academia, its effect, especially when multiple uncertainties occur simultaneously, is poorly understood (Flanagan et al., 2005a). While studies (e.g. Eppinger, 2001) have shown that improvements can be obtained by restructuring the process, the presence of numerous uncertainties obscures such opportunities for improvements. Limited understanding of the impact of different uncertainties on the process presents challenges during design planning.

Product-process interdependencies

Understanding the connectivity between different aspects of the design is not straightforward. Even if one considers only the product domain, the implications of changes to one component on other parts of the design are not easily predicted (Eckert et al., 2004b). In addition, as can be seen from the above models of design, the structure of the product can strongly influence the process. This is particularly true for stable products where the process is derived based on many years of experience and key product dependencies are reflected in the process (e.g. Evans, 1959). However, the knock-on effect on the process, resulting from changes in the product domain, is not always easy to predict, especially in cases where the product architecture is

evolving. Also, some dependencies between tasks are not obvious early on; only as the design matures does it become clear which component configuration will be chosen and hence which task-order is most appropriate. As a result, planning and modelling design projects creates unique challenges.

To deal with these challenges, designers use a host of different documents (e.g. product models, procedural documents, Gantt charts, and Bills of Material) concurrently (Eckert and Clarkson, 2003). Nonetheless, they are forced to rely on their overview of the product and the process to link together information from these different sources. For complex products, however, few people if any have a complete overview: better models and representations are needed to reduce dependency on overview and to improve process efficiency and transparency (Flanagan et al., 2006).

3.2.3 Design planning: a complex-system behaviour-prediction problem

Design planning can be considered a complex-system behaviour-prediction problem because design processes exhibit many characteristics of complex systems including:

- *Feedback loops: If tasks fail to deliver the desired level of information, design iteration may take place.*
- *Non-linear behaviour: many tasks have several possible outcomes such that small changes in the task input information can have a major impact on its outcome.*
- *Uncertainty: task-durations, resource availability, customer requirements and supplier performance are all uncertain.*
- *Emergence: Earl et al. (2004) discuss emergence in the context of helicopter design stating that "the design process may have discernible overall emergent characteristics which may not be predictable from the characteristics of its elements". They argue that designers aim to avoid such behaviour by locating designs within margins.*
- *System resetting: The idea of periodic resetting finds parallels in industrial settings where gateways perform a resetting function after discrete time intervals (Cooper, 1993). Decisions made at such points reduce the uncertainty associated with the project, thus counteracting the effects of time-dependent combinatorial complexity.*

- *Complex adaptive behaviour: Design processes adapt to the context in which they are executed. The process can adapt in response to changes in a supplier's process. This may have new implications for the supplier's process such that it may respond with further adaptation.*

Because design processes exhibit these complex-system characteristics, it is extremely difficult to predict their behaviour – a factor which results in multiple planning challenges.

This section has highlighted the complex nature of design projects and argued that planning such projects is a major challenge. The next section considers research into planning complex processes and concludes with a discussion of the implications of this research for design planning.

3.3 Approaches to analysing and planning complex processes

Doyle (2005) defines planning as: *"Deciding upon a course of action before acting. A plan is a representation of a course of action.... Planning is a problem solving technique. Planning is reasoning about future events in order to verify the existence of a reasonable series of actions to take in order to accomplish a goal."* He also notes that the benefits of planning include *"resolving goal conflicts and providing a basis for error recovery."*

Suchman (1987) discusses the use of plans and situated actions – i.e. unplanned actions triggered by a specific context – to achieve goals. Weld (1994) states that *"Planning is appropriate when a number of actions must be executed in a coherent pattern or when the actions interact in complex ways... Situated action is appropriate when the best action can be easily computed (recognised) from the current state of the world (i.e. no look ahead is necessary because actions do not interfere with each other)."*

During everyday life people use a combination of both approaches; for example, one would frequently plan a trip abroad, pre-booking taxis, train-tickets, flights and hotels while other goals are achieved through situated actions: someone may call at a bank when passing and deposit a cheque, even though this action is not pre-planned – rather it is triggered by the context of passing the bank. In practice, the division between

plans and situated actions is not always clear-cut: some actions that appear to be planned are performed in the context of situational clues.

Both plans and situated actions are also common in the workplace: plans are used to co-ordinate activities in order to ensure that work is completed by a deadline while situated actions encompass unplanned activities such as answering emails and phone calls. Much engineering work involves interdependent actions; in such cases, planning is appropriate.

The next sections review literature on simulation analysis, operations research and AI planning; while design processes exhibit unique characteristics, their planning can nonetheless draw on research from these areas.

3.3.1 Simulation analysis

Computer simulation is increasingly being seen as a powerful way to model and explore the dynamic behaviour of some complex systems (Johnson, 2001). It has been used in numerous fields including engineering, science, management, economics, business and military logistics. Computer simulation is often used when the alternative approaches of direct experimentation or non-stochastic, mathematical modelling are respectively considered impractical or inadequate. Direct experimentation is often unrealistic or impossible because all projects are different while non-stochastic mathematical models cannot adequately capture the dynamic and transient effects encountered in project management problems (Pidd, 1992). Simulation also offers other advantages including improved scalability, control over project variables and richness of output results available. These results, in turn, are easier to access, analyse and interpret than actual project data and are more easily reduced to visual summaries such as histograms and Gantt charts. Such visualisations, complemented by numerical data from the simulation analyses, provide useful insights into the system being examined. An introduction to simulation is given by Ingalls (2002) and a concise review of the topic is provided by Nance and Sargent (2002).

When simulating, elements of the real system are mapped to abstract elements in the simulation model. Execution of the simulation program represents the passing of time during which the state of these elements changes. Simulation research attempts to map this new state back to the real system, arguing that observations of the simulation

behaviour can be used to gain insight into the real system (Zeigler, 1976). A model of the system is used as input into the simulation. Issues concerning simulation input models are discussed by Biller and Nelson (2002) and Chwif et al. (2000) who note that errors in the model are likely to render the simulation analyses invalid.

Nonetheless, simulation is an accepted approach in many fields, especially in modelling of physical systems (Johnson, 2001) where the soundness of results can usually be established using systematic validation techniques (Pidd, 1992 and Sargent, 1998). Indeed, even when the simulation results are incorrect, interrogation of the analysis to uncover the source of errors can prove valuable in highlighting hidden assumptions about the way in which the project is expected to behave. This merit of simulation analysis is particularly relevant in project management.

Simulation in project planning

The value of simulation in project management has been recognised for over forty years. Pidd (1992) provides a general overview of simulation in Management Science domain while Nelson (2004) reviews pivotal simulation research in the Management Science journal. Such reviews, however, cover a broad range of topics related to Management Science and do not provide detailed discussion of the application of simulation for project planning and management in design.

In contrast, a paper by Haga and O'Keefe (2001) discusses the development of PERT/CPM (see section 3.3.2) and associated simulation analyses for exploring the effects of task duration uncertainty. Some of the limitations of PERT/CPM methods, such as failure to model iteration, were addressed by Pritsker and Happ (1966) through the Graphical Evaluation and Review Technique (GERT). Nonetheless, GERT and the later extension, Q-GERT (Pritsker and Sigal, 1983), have not achieved the same acceptance among practitioners as the original PERT/CPM approach.

Despite a relative abundance of general simulation tools (Rizzoli, 2005), commercial tools for project risk analysis are comparatively sparse. Examples include Pertmaster (www.pertmaster.com), Risky Project (<http://www.intaver.com/>) and Risk+ (<http://www.cs-solutions.com/products/?Product=Risk%20Plus>), which capture uncertainty in cost and duration. However, such tools fail to provide any information about the influence of rework, on project duration. This issue has been considered by Cooper (1993), who performed simulation analyses of project plans, using proprietary software, to evaluate the importance of rework discovery time in NPD. The related

issue of iteration was explored by Cho and Eppinger (2005) using a prototype simulation tool.

All of these tools aim to provide insights into projects and many use visual summaries of simulation results (Fig. 3.9) to help project managers identify and reduce project risk. Nonetheless, the industrial application of simulation in design project planning is limited, arguably due to scale of the projects as well as limitations of the tools.

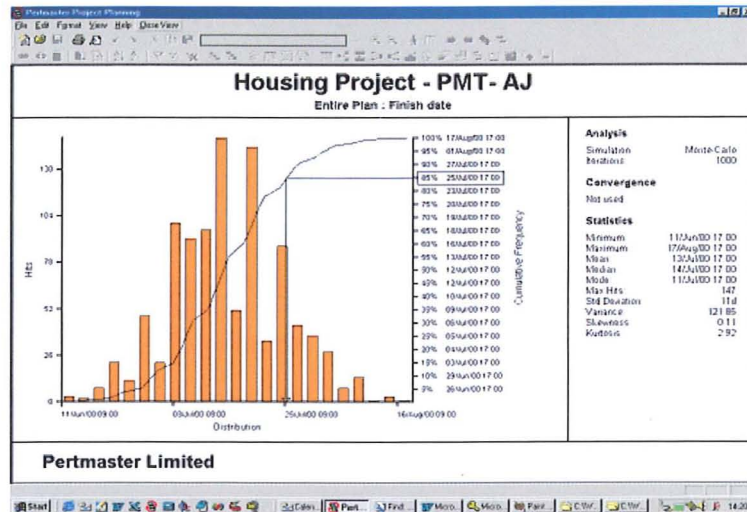


Fig. 3.9 Screen shot of a histogram from Pertmaster (www.pertmaster.com)

3.3.2 Operations research

Operations Research (OR) is a vast research field, so expansive that research into the history of operations research is considered by some as a research-worthy topic in its own right (Gass and Assad, 2005). Operations research can be defined as *"the use of quantitative methods to assist analysts and decision-makers in designing, analysing and improving the performance or operation of systems"* (Carter and Price, 2001: 1). The underlying concepts and methodologies of OR have been developing throughout the history of science and mathematics but the terminology Operations Research is generally considered to have been adopted early in World War II (Hillier and Lieberman, 1974). During this time, there was an urgent need to allocate scarce resources to military operations. This led British and American Military management teams to apply scientific approaches when dealing with this and other strategic and tactical problems, thus creating the first operations research teams. Since then, operations research has been successfully applied in industry, production, military, management and administration (Singh, 1972; Carter and Price, 2001).

As the name suggests, OR involves research into operations within organisations (as opposed to purely theoretical research in a controlled environment). The OR process begins by carefully observing and formulating the problem and then constructing a scientific model which attempts to abstract the essence of the real problem (Hillier and Lieberman, 1974). Such models are then analysed, often using computer programs, and insights thus obtained guide management in decision making, with the goal of improving the efficacy of the organisation by finding optimal solutions.

OR embodies a broad range of tools and techniques, including linear programming, dynamic programming, queuing theory, inventory theory, game theory and simulation, all of which are described in the literature (Hillier and Lieberman, 1974; Carter and Price, 2001). While literature from the OR community into engineering design is sparse (O'Donovan, 2004), the research reported in this thesis has strong parallels with that of OR research: it involves observations of design operations within real organisations and problem formulation (Chapter 4), model-development of a real-world system (Chapter 8) and simulation analysis (Chapters 7 and 8). The interested reader is referred to more-extensive texts (Hillier and Lieberman, 1974; Carter and Price, 2001; Singh, 1972; Gass and Assad, 2005).

Operations research into project planning and management

Extensive reviews of techniques and methods commonly used in project management can be found in the Project Management Body of Knowledge PMBOK (PMI, 2000) or in standard text books such as Kerzner (1992); this section provides only a brief introduction to concepts that are of importance to design planning.

Generic approaches to project planning include CPM, the Critical Path Method, developed by DuPont in 1957 (Kerzner, 1992) and PERT, the Process Evaluation and Review Technique (Malcolm et al., 1959), both of which are instances of the Precedence Diagramming Method (PDM). In CPM, the critical path is the longest sequence of consecutive tasks that establishes the minimum length of time for project-completion; any delays to these tasks will result in project overrun (Horowitz, 1967). In both PERT and CPM, activities are shown as nodes and arrows between activities represent information or material flow (Fig. 3.10). A major difference between both approaches is that CPM uses a modal estimate for task duration, while PERT uses a weighted average of lowest, highest and most likely duration.

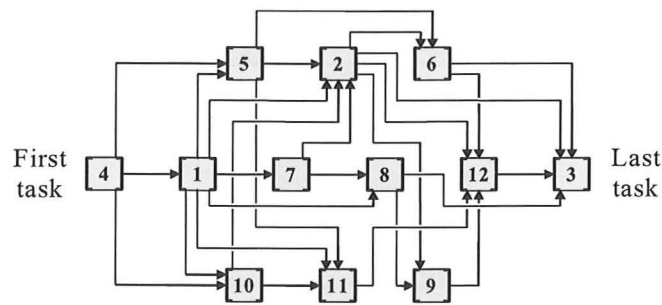


Fig. 3.10 PERT representation of a design process (Clarkson et al., 2000b)

Microsoft Project™ and Primavera™ are among the most commonly used software implementations of PERT and CPM. While these tools show task ordering and can highlight the knock-on effects of changes to a given task, they suffer from the inherent limitations of PDM: at the outset of a design project many precedences are unknown or uncertain; in industry the existence of precedence relationships often becomes clear only when tasks are executed. Neither iteration nor task alternatives can be modelled using PDM, a limitation which detracts from their value in modelling and improving design processes. Another criticism of current tools is that they offer no information on the reason(s) for precedence between tasks. Often, knowing the nature of the precedence relationships is as important as being aware that a relationship exists, but industrial software tools fail to provide this information.

In addition to software tools, extensive project management methodologies, such as PRINCE2 (PROjects IN CONTROLLED ENVIRONMENTS), shown in Fig. 3.11, have been developed in order to improve project planning and control (www.prince2.com). The PRINCE2 methodology was developed by the UK Central Computer and Telecommunications Agency (<http://www.ccta.gov.uk/>) and highlights the benefits of planning in achieving the desired level of product quality on time and on budget, subject to a given set of preconditions and assumptions. It also notes the importance of re-planning as the project unfolds, contingency planning and scalability of plans – for large projects, a hierarchy of plans is advocated. Finally, it recommends that planning and risk analysis should go hand in hand and stresses that plans should be used to co-ordinate activities and resources, improve communications and provide a mechanism for project control. The arrows denote the chronological order in which different factors are considered. While PRINCE2 is not aimed specifically at engineering design processes – its original application was telecommunications – all of these issues are pertinent.

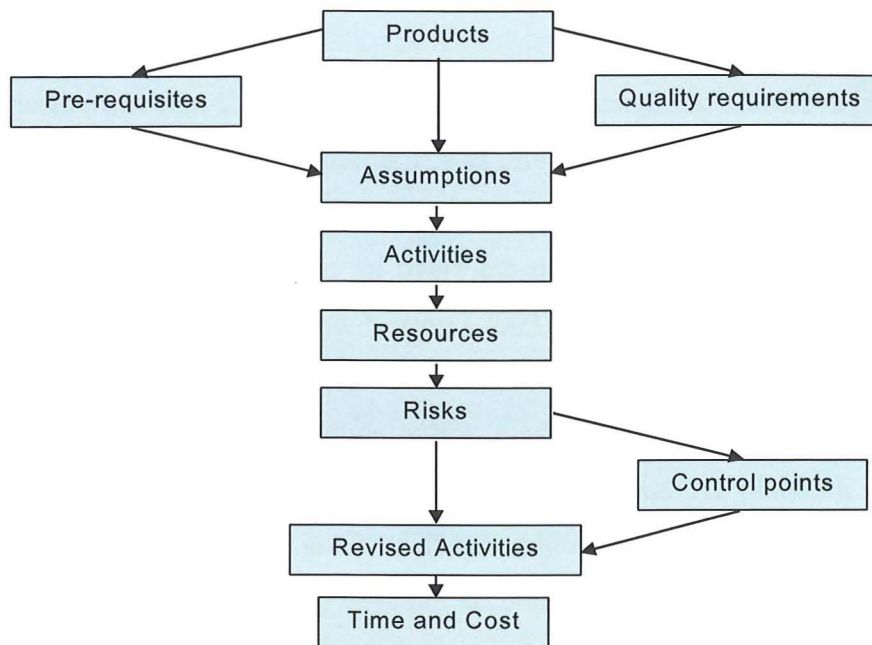


Fig. 3.11 The components of a plan (www.prince2.com)

Implications of OR research on design planning

Despite the importance of design planning in achieving on-time project completion, the literature on design planning practice is surprisingly sparse. Eckert and Clarkson (2003) argue that large-scale project planning in industry is very complicated, that designers do not always follow plans and instead resort to fire-fighting, especially when under extreme time-pressure, that work arises which is not captured by plans, and that understanding, managing and updating the connections between different plans is difficult and error-prone.

The high uncertainty associated with design activities limits the suitability of techniques such as PERT/CPM for design planning: due to uncertainty, it is difficult to predict which tasks will be critical at the outset of the design process. Nonetheless, the fundamental concept of operations research – observing real-world operations, modelling them and trying to improve them is fundamental to this research. The concept of simulation, as a means to explore design processes, is also appealing: their complex nature restricts the ability of alternative techniques to provide useful insights. Finally, the literature on project management research highlights several practical challenges of planning and managing large-scale projects, which are overlooked by abstract process models. Thus, literature from the operations research domain provides useful insights into tools and techniques aimed at overcoming the practical challenges of large-scale project planning.

3.3.3 AI planning

In parallel to literature on simulation and OR, much research has also been performed into Artificial Intelligence (AI) planning. It was this community who defined planning as a 'wicked problem', a phrase which is also used to describe design (Rittel and Webber, 1984). This section gives an overview of AI planning. A more complete review can be found in Russel and Norvig (2003), Callan (2003) or Charniak and McDermott (1987).

Applications of AI planning include robot motion planning, military planning, logistics planning and manufacturing assembly scheduling. AI planning problems require well-defined tasks and goals, a factor which can limit their suitability to design planning – pure AI planning does not account for gaining information from the environment which could lead to changes in the plan (Callan, 2003). Also, AI planning involves either searching the state-space or the plan-space (space of partially ordered plans), which leads to problems when the number of possible states becomes too large. Nonetheless, some parallels between both domains (AI and design planning) exist and a brief review of relevant literature was considered appropriate.

Overview of AI planning

In AI, a planning domain is made up of a set of operators or action types, an initial state and a goal-state(s). Execution of an operator is only valid in some particular set of world states (its preconditions), and each execution has a specific set of effects on the world state. A planning problem is solved by producing a sequence of actions, which when executed in a world satisfying the initial state description, will achieve the goal.

The General Problem Solver (GPS) (Newell and Simon, 1963) was one of the first programs that tried to solve planning problems usually requiring human intelligence. It introduced means-end analysis as its search. It tried to determine the difference between the current and goal-states in order to identify actions that minimise that difference. As the approach focuses on the goal-state, it combines aspects of forward and backward search and is an improvement on brute force approaches. Unfortunately, GPS can only solve highly structured problems and is best suited to problems such as propositional calculus proofs.

The following section describes different approaches to AI planning and is followed by a discussion of resulting implications for planning in engineering design.

Approaches to AI planning

Broadly speaking, AI approaches to planning can be grouped into five categories: non-hierarchical, hierarchical, case-based, opportunistic and partial order planning. The basic concept of each category is described below. Some planners fit more than one category.

Non-hierarchical planning: Non-hierarchical planners are planners which do not prioritise goals – all goals are treated with equal attention. A famous example of non-hierarchical planning is the Stanford Research Institute Problem Solver (STRIPS) (Fikes and Nilsson, 1971, as described in Russel and Norvig, 2003). The effects of STRIPS operators are defined in terms of three groups of expressions: preconditions, deletions and additions. Preconditions describe the conditions that must be true in order to apply an operator. The deletions consist of a set of expressions that must be deleted from a model of a situation if the operator is applied. Conversely, the additions comprise a set of expressions that must be added to a model of the situation if the operator is applied. STRIPS combines backward search and means-end analysis to provide a reasonably fast means of solving planning problems. One limitation of STRIPS is its lack of expressiveness, but it has proved useful in solving some real world AI problems such as robot-motion planning (Russel and Norvig, 2003).

In addition to being non-hierarchical, STRIPS is also an example of a linear planner because the order in which sub-problems are solved is linearly related to the order in which the actions of the plan are executed. Non-hierarchical, non-linear planners also exist (e.g. SIPE and TWEAK). Applications of such planners include theorem-proving applications, robot-motion planning, toy problems and puzzles.

Hierarchical planning: Because non-hierarchical planners are unable to prioritise goals, they sometimes become overly focused on unimportant details. To address this concern, hierarchical planners begin by creating vague plans that satisfy the most important goals. They split up a planning domain in different levels of abstraction, each with increasing level of detail. Actions are then planned first at more abstract levels (e.g. go home from work). These high-level actions are then described in terms of lower-level actions (e.g. take the bus). Hierarchical planners proceed by computing plans at a high level of abstraction and then incrementally refining these plans, by descending in the abstraction hierarchy, until a fully specified plan is realised.

The first example of a hierarchical linear planner was GPS (described above). A more versatile approach, ABSTRIPS, was developed which builds on the principals of STRIPS, but plans in a hierarchy of abstraction spaces. The use of hierarchy permits the early identification of dead-ends and avoided pointless search. One problem with ABSTRIPS is that it does not provide information on sub-plan failure within the context of the overall plan.

Hierarchical non-linear planners have also been developed, the most famous of which is NOAH (Nets Of Action Hierarchies). One application of NOAH involved defining the steps in a repair task where it could provide abstract guidance for an expert and more detailed information for a novice. It also had features to resolve conflicting interactions and redundancies in the plan. NOAH is an example of a least commitment planner as it does not order sub-goals until necessary (Russel and Norvig, 2003). Another example of a hierarchical, non-linear planner is SIPE2. Such planners have applications in military operations planning, construction planning and production-line scheduling, (<http://www.ai.sri.com/~sipe/>).

Case-based planning: Case-based planners work on the premise that very few completely new plans exist. Conversely, new instantiations of existing plans are common: old plans are refined to transform them into new, useful plans. The challenge lies in recognising which plans can be reused. One suitable application of case-based planning is the design of experiments where previous plans provide a useful starting point. Case-based planning offers a potential mechanism for handling intractable problems because it is considerably faster than planning from scratch. An example of the case-based planning approach is CaPER (Kettle et al., 1994) which has been used for transportation logistics planning, particularly in military applications. Spalazzi (2001) provides a more comprehensive review of case-based planning.

Opportunistic planning: Opportunistic planners try to mimic the manner in which humans combine plans and situated actions (Doyle, 2005). Such planners develop plans based on opportunities as they arise. In contrast to hierarchical planners which take a top-down approach, opportunistic planners such as OPM (Opportunistic Planning Model) essentially implement a bottom-up strategy. Suggestions for partial plans are made by specialists and a plan “grows out” of constituent sub-plans; the planner attempts to connect the different sub-plans together in an iterative manner. Opportunistic planners have been applied to real-world applications such as the

reactive routing and scheduling of a fleet of taxi vehicles in response to dynamically changing transportation demands (due to weather, congestion, accidents etc).

Partial-order planning: Partial-order planning attempts to increase planning efficiency by avoiding premature commitments to a particular order for achieving sub-goals (Callan, 2003). The approach delays decisions concerning step-ordering with the plan for as long as possible, as opposed to other approaches which manipulate ordered sequences of actions. Barrett and Weld (1994) compared performance of total and partial ordered planners and found no cases where a total-order planner performed significantly better than a partial-order planner but several domains in which the partial-order planner was exponentially better. They also identified the characteristics of problem domains in which partial-order planners are most effective. A classic example of partial-order planning is NOAH (see above) and most planners since NOAH have been partial-order planners.

Implications of AI planning on design project planning

Most AI planners are based on assumptions (Callan, 2003; Charniak and McDermott, 1987; Doyle, 2005; Russel and Norvig, 2003) which limit their applicability to planning design projects. For example, they assume that goals are fully defined and remain fixed over the course of the plan while in design, requirements frequently change. They also assume that the AI planner has the necessary information to completely model the problem space which is rarely, if ever, true in the context of design, and that the problem-world is only changed by the actions of the agent. In design planning, numerous factors can change the decision making context.

Nonetheless, some ideas from the AI planning community are highly relevant to design planning. The approach of creating vague plans and filling in the details later – as followed by hierarchical planners – echoes with planning during new product development programs where parts of the plan, such as testing, are reasonably well defined from the outset, while other parts of the plan are only known at a low level of detail. In addition, the concept of only ordering sub-goals when necessary reflects the desire in real-world design planning to keep plans flexible in order to better respond to the changing information state of the real world. Further, the problem sub-goal interaction, where the solutions to two separate sub-goals must be interleaved in order to solve them both (e.g. the Sussman anomaly, 1975) finds parallels with the design of interdependent components which often drives iteration.

The notion of reusing parts of plans, as with case-based planning, also echoes with engineering design where partial plans are frequently reused for different projects. In some cases, Gantt charts from previous projects provide the foundations for a new project plan, the latter derived by modification and refinement of the former. Likewise, process documents and workflow instructions define the procedures which must be performed in order to reach the predefined sub-goals. Such procedures may be reused during project iteration or when similarity arises between different plans.

While the literature on AI planning is highly divergent from that associated with OR and simulation, all three domains are united by the common need for an appropriate model of the system being analysed. The next section considers models used to represent and explore design processes.

3.4 Models used to plan design processes

A model is defined as *"a simplified or idealized description or conception of a particular system, situation, or process, often in mathematical terms, that is put forward as a basis for theoretical or empirical understanding, or for calculations, predictions, etc.; a conceptual or mental representation of something"* (Simpson and Weiner, 1989). Further, models are an abstraction of the processes that they aim to represent and therefore only capture a subset of the factors which may influence process-behaviour. Thus models, including those used to model product development processes, are not intended to be exact representations of a system; rather they should characterise a system sufficiently well such that useful inferences can be made. This sentiment was captured by George Box, who stated, *"All models are wrong – but some models are useful"* (Browning, 2004). Also, depending on the purpose of the model, some models may be less wrong than others.

Browning (2004) notes that *"a useful model is helpful for making predictions and testing hypotheses about the effects of certain actions in the real world, where such actions would be too disruptive or costly to try. A useful model also provides insights which are otherwise available only through (sometimes painful) experience."* Further, both descriptive and predictive models are common; while the former can provide useful insights into how a system works, the latter are especially relevant to project planning.

Predictive models are created at a point when the actual process does not yet exist: for example a predictive model for an automotive design process will be constructed before performing the actual design work. The way a project is described and modelled profoundly affects the insights that can be gained and in turn influences the way that the project is carried out.

3.4.1 DSM

The term Design Structure Matrix (DSM) – alternatively Dependency Structure Matrix – was first used by Steward (1981) in his work on the relationships between tasks in processes. Matrices, which illuminate the structures of organisations, products, processes, etc., have existed for many years under a variety of names (e.g. dependency maps, interaction matrices, and precedence matrices). In recent years, DSMs have been successfully used in numerous areas by both academics and industry (Browning, 2001).

DSMs consist of square matrices with identically labelled rows and columns and use off-diagonal entries (tick-marks) to signify the dependency of one element on another (Fig. 3.12). Such matrices can be used to model processes, products or organisations. When used to model the design process, the matrices show task dependencies and can be reordered to achieve minimum iteration. Product models can be generated to show connectivity between different components and organisational DSMs show connections between teams. The benefit of DSMs is that they clearly display relationships between elements of a system in a way that is easy to comprehend and supports analysis of the system. However, they do not display the nature of this connectivity.

Numerous researchers have used DSMs as a basis for their work in modelling the different stages of the design process. The ticks can be replaced by numerical values to show the weighting of relationships. Probability and impact versions of a numerical DSM were introduced to estimate product development time by Carrascosa et al. (1998). They define probability as *“the likelihood of the default value changing over time”* whilst impact was considered as *“the effect of a change on the tasks receiving the information.”*

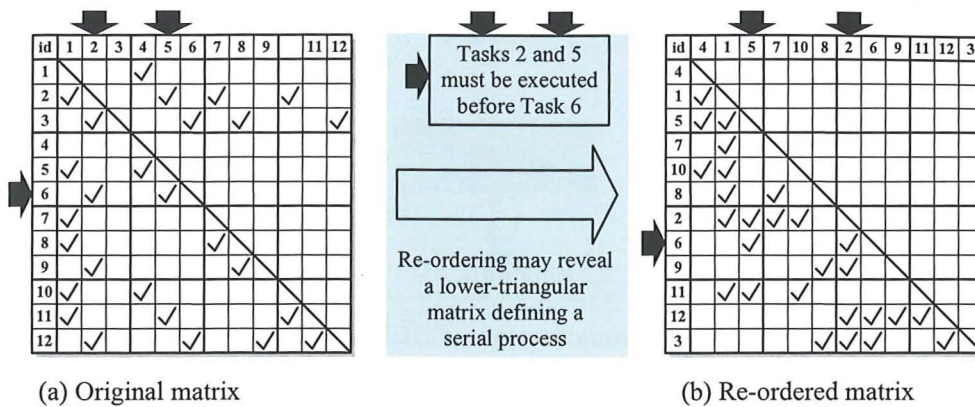


Fig. 3.12 A time-based DSM with tick-marks to denote dependencies (Clarkson et al., 2000a)

More complex analysis has been attempted using DSMs through Structural Sensitivity Analysis (SSA), which is based upon the concepts of Sensitivity and Variability (Yassine et al., 1999). Sensitivity refers to the relative degree of changes in a task and its predecessor, whilst variability is a measure of the problems in creating reliable estimates for use in the task.

Despite their utility as a process representation, conventional DSMs fail to show the reason for dependencies between tasks and yield no information on task durations. Also, while they can be used to highlight groups of independent tasks, they provide little guidance on the most suitable execution order for these tasks. In light of these limitations, other design process modelling frameworks were also considered for this research.

3.4.2 IDEF

The U.S. Air Force programme for Integrated Computer Aided Manufacturing (ICAM) resulted in the Integration Definition for Function Modelling (IDEF) techniques which are based on the Structured Analysis and Design Technique (SADT) developed by Softech Inc. (1981). Two members of the IDEF family of models, IDEF0 and IDEF3 are relevant to process modelling.

IDEF0 represents a process as being composed of a sequence of activities, each having inputs, controls (e.g. standard working practices), outputs and mechanisms (e.g. people, tools) as shown in Figure 3.13. IDEF0 includes both a comprehensive methodology for developing models and a definition of a graphical modelling language. It can be used to construct structured representations of functions, activities or processes within the modelled system or subject area. Currently, IDEF0 supports modelling efforts for a wide range of applications (Marca and McGowan, 1993).

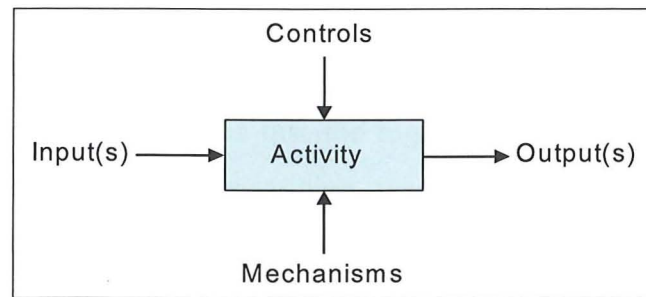


Fig. 3.13 IDEF0 representation of a task

In contrast to IDEF0, which characterises activities in terms of inputs, outputs, controls and mechanisms, IDEF3 – the Process Description Capture Method – represents activities by characterising the timing and sequencing of critical process events (Meyer et al, 1995; Kim et al., 2001). It models the dynamic behaviour of a process using flow diagrams which indicate both temporal and information-based precedences, and can also capture iteration (Noran, 2000). Process Flow Diagrams (PFDs) represent conditions of the order in which activities are performed and are constructed from four types of element: units of behaviour (UOB), junctions, precedence links and referents as shown in Figure 3.14 (Kim et al., 2001).

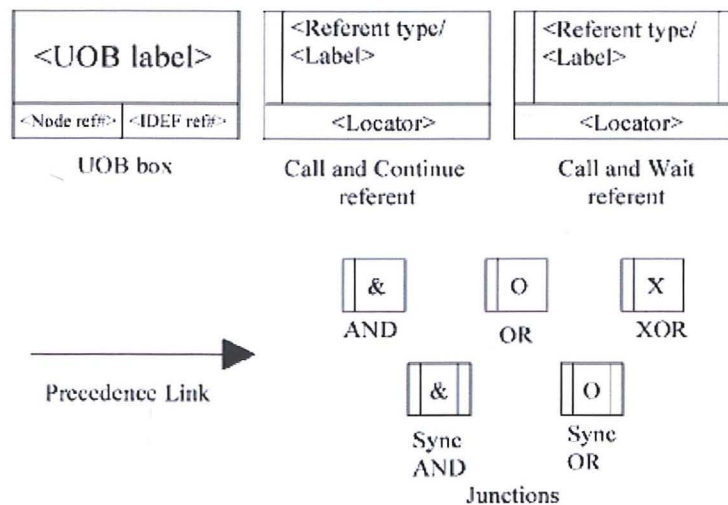


Fig. 3.14 IDEF3 representation of a task

A UOB is characterised in terms of a label (e.g. the task name), intervals of time over which it occurs and the temporal relationships it has with respect to other processes. Precedence links between UOBs represent constraints on process flow. In addition to precedences, junctions are used to show the process logic. They specify process routes at 'branches' in the process by defining time and logical relationships such as AND, OR and XOR. Referents are used to represent additional information related to a

UOB. They allow the modelling of process iteration and aim to enhance understanding, provide additional meaning, and simplify model construction.

One drawback of IDEF models is that the high level of detail required results in problems when creating and maintaining models. Also, they can prove overly restrictive when modelling uncertainty and iteration in design, problems which do not arise when using the Signposting model described below.

3.4.3 Signposting

Signposting is a dynamic design process model based on task connectivity due to parameters (Clarkson and Hamilton, 2000). Output parameters from one task are used as inputs to another. The state of a parameter is indicated in terms of the confidence that the designer has in its refinement; a set of parameter states defines the design state. A task-order is implicit in the confidence values and the effect that one task has on the process is determined by confidence mapping (Fig. 3.15). Iteration can be modelled to a limited extent by re-running a task with different confidence values.

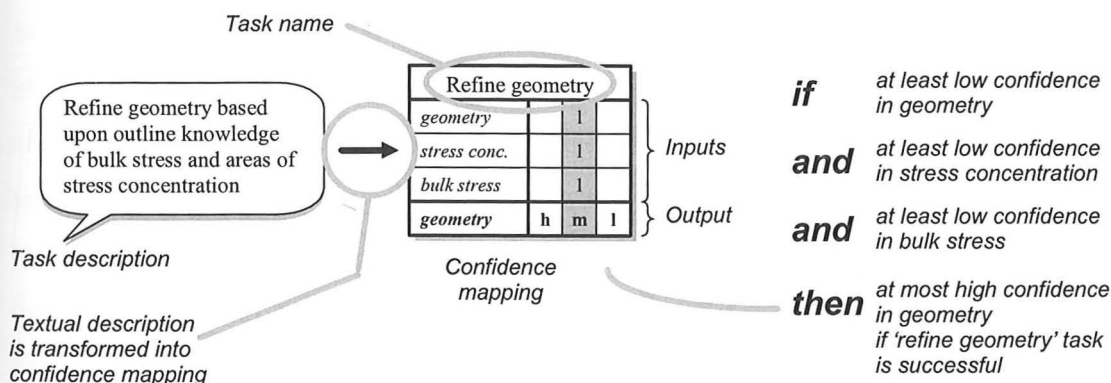


Fig. 3.15 An example confidence mapping (Clarkson et al., 2000b)

Initially the Signposting approach was used to guide designers to the next task, by highlighting those tasks for which they had sufficient input data. Later, the technique was developed to support optimum task ordering by selecting the most appropriate option from a list of available tasks (Melo and Clarkson, 2001; Jarrett and Clarkson, 2001). Figure 3.16 shows the routes through a design process, generated using the Signposting model. Markov chains were used to establish the best policy (preferred task-order) in terms of cost and risk to reach the design goal (Melo and Clarkson, 2001). The use of Markov chains provided insight into hidden task precedences for a serial design process which become important in the event of rework (Section 5.3.1);

its application to concurrent, resource-limited implementation of the Signposting model is, however, problematic.

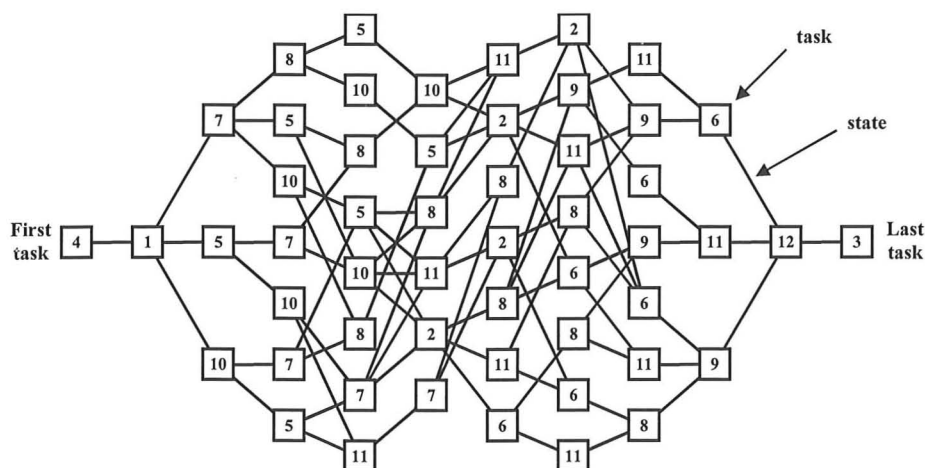


Fig. 3.16 Signposting task-sequence diagram (Clarkson et al., 2000a)

A project simulation tool, based on the Signposting approach was developed by O'Donovan (2004), who also extended the Signposting model to include such features as resource constraints and learning during rework. The extensions allowed the modelling of non-Markov processes. To realise this increased functionality, parameters are assigned numerical levels that are required by and/or altered by tasks, an approach which diverged slightly from the original confidence concept, but nonetheless indicates the maturity and context of parameter-information within the design process. The extended Signposting framework can be used to construct detailed models of design processes – it can capture multiple possible outcomes (different degrees of success, different modes of failure) and estimates of their likelihood, different task inputs (information, workforce and other resources) and their impact on the outcomes possible, and estimates of the duration and cost of different iterations (O'Donovan et al., 2004; Fig. 3.17). The provision of primarily graphical support for Signposting-model elicitation and representation, in the context of aero-engine development, has been considered by Wynn, based on findings from an 8-month on-site industrial study. His work has focused on capturing information-driven dependencies and iteration in design processes and using sophisticated hierarchical structures of tasks and parameters to support manipulation and presentation of models (Wynn et al., 2006).

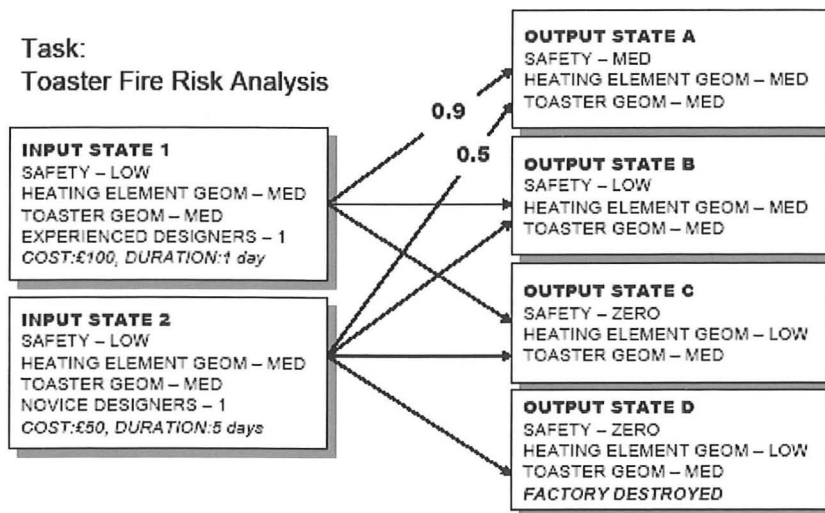


Figure 3.17 Modelling resource information (O'Donovan, 2004)

As with the other modelling frameworks considered, however, Signposting is not without its limitations. Problems arise due to different interpretations of the meaning of confidence (Section 5.4.1), when modelling parallel tasks that affect the same parameter and when modelling task precedences (Section 5.1.3). Signposting model building and representation can also prove challenging due to the richness of model data.

3.4.4 Other modelling frameworks

The above models are aimed specifically at design process modelling. However, more generic modelling techniques have also been successfully applied to design (see Browning and Ramasesh, 2005, for a review of models). In particular, implementations of PDM, such as MS Project™ and Primavera™ are commonplace in industry. Limitations of PDM, when used to model design processes, were discussed in section 3.3.2.

Likewise, Petri nets have been successfully applied to design, in addition to applications in other areas such software development and manufacturing. Petri nets are comprised of places, transitions, arcs and tokens (Fig. 3.18). Places generally refer to process inputs and outputs and transitions are the possible actions (tasks in the case of design processes). Arcs connect places to transitions, indicating the inputs required for a transition and the resulting outputs. Finally, tokens represent the presence or absence of objects represented by different inputs/outputs. As transitions occur, tokens move from input to output places. Some authors have directly applied Petri

nets to engineering problems (Dou and Cai, 2002) while others have constructed design-specific modelling frameworks within the broader context of Petri nets (McMahon et al., 1993; Horvath et al., 2000). As with IDEF, however, the use of Petri models can prove overly restrictive for applications in design and challenges arise when building and visualising large models. Difficulties may also occur when modelling multiple instances of the same task in a different context – e.g. during design iteration.

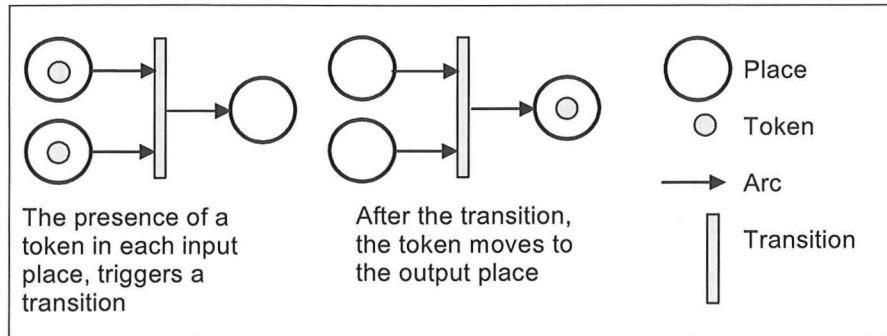


Fig. 3.18 Diagram of a Petri net

Another generic approach, which has applications in supporting design projects, is the use of workflow systems. Workflow systems help automate business or laboratory processes, in which documents, information or tasks are passed from one participant (machine or human) to another for action, according to a set of procedural rules (Flattery, 2005). Workflow systems combine influences from a variety of disciplines, including cooperative information systems, computer-supported cooperative work, groupware systems, active databases, and planning (Allen, 2000). Such systems help companies co-ordinate information and ensure that the right information gets to the right people quickly. Hence, their utilisation can reduce errors and expedite processes. On the downside, however, workflow systems can reduce process flexibility, lead to over-management and decreased employee empowerment, and can prove costly to implement and maintain (CTG, 1997).

3.4.5 Comparing the different models

This research aimed to provide support for design-planning. In order to do so however, it was necessary to select a modelling framework which can be used to capture and analyse processes. This task was non-trivial because different models have different merits and limitations.

IDEF models (both IDEF0 and IDEF3) have the advantage that they include a comprehensive methodology for model-development and that they can be scaled to capture large processes. A drawback of such models, however, is that they can prove overly-rigid and hence are better suited to capturing stable, repeatable processes than to modelling processes rich in uncertainty. In particular, the absence of a means to model iteration in IDEF0 is a clear disadvantage. Due to these factors, IDEF was considered unsuitable for this research.

In contrast to IDEF0, DSMs capture and represent iteration effectively. Another advantage of DSMs is their comparative simplicity, both during model capture and analysis. However, the simplicity of DSMs as a modelling approach can also prove to be a drawback – they fail to model data about alternative process routes, a limitation which can cause problems when considering alternative design approaches. In addition, conventional DSMs do not show how tasks are connected – they fail to capture design dependencies at a parameter-level, a factor which limits their applicability when examining the product-process link. These issues meant that a completely DSM-based approach was not ideally suited to this research; notwithstanding that DSM is a highly valuable representation of the design process.

Other models too were considered. PDM models are frequently applied in industrial practice – despite their inability to model rework, they constitute a concise representation of task timing as well as an impression of process connectivity. Their failure to capture iteration and uncertainty does, however, limit their suitability to this work. Likewise, Petri nets can act as a valuable means of visualising complex processes, although their utility diminishes when the processes become very large. In addition, building a Petri-net model of a large-scale design process is likely to prove difficult.

In light of the findings from the complexity literature (Section 3.1) and the characteristics of design process (Section 3.2.2), three criteria were considered especially relevant for this research: 1) ability to capture rework, 2) capability to explore sensitivity to different uncertainties and 3) capability to analyse the product-process link. Signposting was chosen for this research due to a combination of factors:

- Firstly, Signposting can model rework in great detail. While other modelling frameworks such as DSMs can also capture rework, unlike Signposting they

IDEF models (both IDEF0 and IDEF3) have the advantage that they include a comprehensive methodology for model-development and that they can be scaled to capture large processes. A drawback of such models, however, is that they can prove overly-rigid and hence are better suited to capturing stable, repeatable processes than to modelling processes rich in uncertainty. In particular, the absence of a means to model iteration in IDEF0 is a clear disadvantage. Due to these factors, IDEF was considered unsuitable for this research.

In contrast to IDEF0, DSMs capture and represent iteration effectively. Another advantage of DSMs is their comparative simplicity, both during model capture and analysis. However, the simplicity of DSMs as a modelling approach can also prove to be a drawback – they fail to model data about alternative process routes, a limitation which can cause problems when considering alternative design approaches. In addition, conventional DSMs do not show how tasks are connected – they fail to capture design dependencies at a parameter-level, a factor which limits their applicability when examining the product-process link. These issues meant that a completely DSM-based approach was not ideally suited to this research; notwithstanding that DSM is a highly valuable representation of the design process.

Other models too were considered. PDM models are frequently applied in industrial practice – despite their inability to model rework, they constitute a concise representation of task timing as well as an impression of process connectivity. Their failure to capture iteration and uncertainty does, however, limit their suitability to this work. Likewise, Petri nets can act as a valuable means of visualising complex processes, although their utility diminishes when the processes become very large. In addition, building a Petri-net model of a large-scale design process is likely to prove difficult.

In light of the findings from the complexity literature (Section 3.1) and the characteristics of design process (Section 3.2.2), three criteria were considered especially relevant for this research: 1) ability to capture rework, 2) capability to explore sensitivity to different uncertainties and 3) capability to analyse the product-process link. Signposting was chosen for this research due to a combination of factors:

- Firstly, Signposting can model rework in great detail. While other modelling frameworks such as DSMs can also capture rework, unlike Signposting they

cannot account for alternative task-failure-outcomes or dynamic task re-ordering during rework.

- Secondly, Signposting models are rich in data and can capture numerous sources of design process uncertainty. Thus, Signposting was considered an appropriate approach for exploring sensitivity to different uncertainties in complex processes.
- Thirdly, Signposting can be used to model the product-process link. The notion of parameter confidences, which lies at the core of Signposting, has the potential to capture the effect of different tasks in the process on product confidence at a parameter level.
- Finally, the Signposting model was developed in the EDC at Cambridge and much tacit knowledge concerning the model is available from more experienced group members.

Although Signposting was considered most appropriate for this research, it was noted that other models (particularly DSM) had advantages in some applications and that opportunities existed to improve the Signposting model, particularly in terms of providing support for modelling and representation. Enhancements to Signposting are discussed in Chapter 6.

3.5 Chapter reflection: opportunities to support design planning

Design processes exhibit complex systems characteristics such as feedback loops, non-linear and complex adaptive responses, uncertainty and emergence. As a result, their behaviour is difficult to predict. In particular, the impact of rework and product-process interdependencies warrant attention. Likewise, a means to determine sensitivity to different uncertainties would prove useful. Sensitivity analysis capability could also be effectively deployed to investigate process variations in response to scale and connectivity level.

Different research communities (OR, AI and simulation) provide useful techniques for complex system behaviour prediction. The successful application of such techniques within the context of engineering design, however, is dependent on the appropriate model-choice: different models have different merits and limitations. In light of the challenges of modelling and analysing complex systems and the characteristics of

design processes, Signposting was considered the most appropriate model for this research. Nonetheless opportunities to improve the Signposting modelling framework and simulation analysis tool exist.

In summary, this chapter has reviewed existing academic work relevant to design planning support and hence identified research opportunities which focus on:

- *Modelling and representation;*
- *Rework;*
- *Product-process interdependencies;*
- *Sensitivity analysis to uncertainty and process variations.*

Opportunities to support design planning, based on an industrial case study, are the topic of the next chapter.

4

Project planning: an industrial case study

Despite the relatively large body of relevant literature from related domains, academic research which aims to understand the practical challenges of design planning in industry is surprisingly sparse (see Eckert and Clarkson, 2003, for a rare example). This research aims to bridge this gap and critique information obtained from the literature review. It sets out to complement the broad overview of design processes presented in the previous chapter, by providing deeper insights into planning practice within a single company. The core findings of this chapter motivate the remaining chapters of this thesis.

The case study objectives were manifold and included the following goals:

- 1) *To identify the planning challenge(s) of planning large-scale projects.*
- 2) *To find out how projects are planned;*
- 3) *To identify the users and uses of plans;*
- 4) *To establish how projects succeed despite their complexity.*

In pursuing these goals, the case study aimed to shed light on the core topics of this thesis: modelling, rework, scale and connectivity, and the product-process link.

Before proceeding to these issues, however, it is important to understand the context of the case-study company in so far as this affects the generalisability of the research findings. The initial sections of the chapter introduce Perkins engines and characterise the company's design context.

4.1 Perkins Engines: company description

The Perkins organisation was founded by the late Mr. Frank Perkins in 1932. The company, originally based in Queen Street, Peterborough, began by producing the Vixen engine, and later the Fox, Wolf, and Leopard series of engines. The first six cylinder engines were manufactured in 1937 and found widespread applications in agriculture. During World War II, Perkins produced engines for military use, including marine engines for naval rescue craft.

In 1959, the company was purchased by Massey Ferguson (a tractor manufacturer, which is currently a member of the AGCO Corporation). Since then Perkins has been owned by Rolls Royce and Lucas Varity, before being purchased by Caterpillar in 1998.

In 1988, the engineering division of Perkins decided to take advantage of the company's excellent reputation for quality and to offer its capabilities for external business, under the name Perkins Technology Consultancy, or PTeC for short. Since its establishment, PTeC has performed many projects on a wide range of diesel engines for its customers.

Perkins has grown to become a world leader in the design and manufacture of off-highway diesels, producing over 300,000 engines per year. It is a world-wide company with large manufacturing sites in the UK and Brazil, supplying more than 1000 customers in over 160 countries. Key customers include Massey Ferguson, Landini, Volvo, JCB and Caterpillar.

The Perkins product range includes both diesel and gas engines for the 4 kW - 2,000 kW market. These engines are primarily used in off-highway diesel applications (agriculture, construction) and electricity generation. Perkins maintains a strong position in its key markets; one in five tractor engines worldwide, for example, are supplied by Perkins. Further details can be found at the company's website (www.Perkins.com).

4.2 Characterisation of Perkins design context

While Perkins is currently successful at delivering new products, it nonetheless faces several challenges due to changes in employee age demographics, increasing process concurrency and rising product complexity driven by emissions legislation and market pressure. Many experienced employees are retiring and being replaced with younger designers and contractors who cannot refer back to previous projects. In addition, the complexity of diesel engines is increasing, as is that of their design process. One objective of the current engine-development project at Perkins is to design a new engine in half the time of previous development programmes and for half the cost, despite higher performance criteria. This section describes how diesel engine design complexity is likely to increase and outlines the medium term challenges facing the company. Although the below issues were observed within Perkins, many are also relevant to other engineering companies (Section 4.5).

4.2.1 *Increasing product complexity*

Ever-tougher requirements concerning emissions legislation are driving technical innovation in diesel engines. In particular, new electronics are being introduced to reduce the amount of NO_x emissions. Higher burn-temperatures are also necessary, requiring the introduction of new materials. As diesel engines are a mature technology, many of the simpler options for reducing emissions have already been adopted and further improvements frequently require clever design work.

Perkins is trying to share components with Caterpillar engines in an effort to reduce cost and duplication (the current use of platform components is only 4%). The use of platform components introduces a new level of complexity as trade-offs are required across a wider range of engines – changes to an engine component cannot be made without considering other engines that could be affected.

Perkins produces multiple variants of the same engine in order to meet customer-specific needs (Fig. 4.1). As a result, an ever-wider range of applications must be considered for a given design. For example, one customer may desire to have all of the filters on one side of the engine in order to reduce maintenance while this configuration may not work for another. Allowing wide product variation without introducing high additional costs is difficult.

New technologies and increasing product complexity present a challenge to designers and engineers at several stages in the design process, as trusted technologies are replaced with novel design concepts and materials. Uncertainties are introduced in component procurement lead-time, concept evaluation, design and development, and testing and validation. These uncertainties frequently introduce design process challenges – planners and managers have difficulty in predicting task-duration, task dependency and resource demands for novel design work. As such, changes in product complexity are likely to affect the plannability of projects as discussed in section 4.4.4.

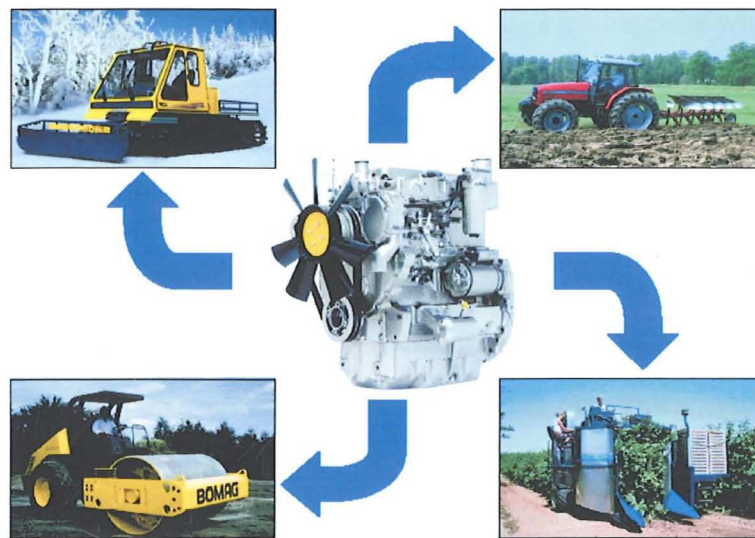


Fig. 4.1 One engine can be used in numerous applications (Jarratt, 2004)

4.2.2 New process risks

Product development times are decreasing due to commercial and legislation demands and this trend is set to continue (Jarratt et al., 2003). As a result, task concurrency-level is increasing and tasks are now being performed earlier in the design process where uncertainties in the task-input information state are higher and the chances of success are reduced. For example, control software may be designed based on synthesised data from assumptions instead of waiting for the engine to reach a high level of completion such that actual performance data can be collected. If these assumptions are incorrect, redesign will be necessary.

The management/administrative effort associated with concurrent design is also higher. If one designer performs three interdependent tasks in sequence, the necessary information exchange is vastly different from three designers performing the same

three tasks concurrently. In the latter case, the likelihood that task failures will creep into the system is increased, leading to rework and iteration. Conventional plans, such as those based on PDM, which assume that all tasks succeed, are likely to prove problematic in this design context.

Sensitivity to failure in predicting task durations increases as greater numbers of tasks become closer to the critical path. The impact of task failures in terms of project duration is likely to rise, as the chances that reworked tasks will be on the critical path increases. It is also likely that failure probabilities for individual tasks will increase if the time available for their completion is reduced. Further, the effects of increasing resources may diminish in highly concurrent designs. This happens when information rather than resource dependencies become critical in determining which tasks can be executed. Under such circumstances, increasing resource availability will not reduce project duration.

4.2.3 Less experienced people

The demographics of the Perkins workforce have undergone major changes in the last few years. Older designers and managers are retiring and much tacit knowledge is leaving the company with them. At the same time, increasing numbers of contractors have been hired, many of whom have a poor understanding of the organisation, its culture and, in some cases, the product.

The related problem of high staff-turnover was particularly relevant in the project office, where a central team of plan-administrators were located. During the 30 months of collaboration with Perkins, this team of six was completely replaced. As a result, much tacit knowledge about planning was lost. Staff-turnover within the design teams was not as severe but nonetheless posed some challenges to the company, particularly with respect to decreasing overview. On the positive side, these changes led to increased competence in computer aided design.

Another issue which relates to changing staff demographics is changes to the organisational structure. As the employees change roles within the company, it becomes difficult for others to track down the experts on a particular issue, because organisational charts do not provide any hints concerning the level of experience that different employees possess. This acts as a barrier to efficient communication even though a positive attitude towards communication prevails within the company.

Despite being very busy, employees find time to answer questions from others during the routine course of their work. Such face-to-face communication is supported by numerous constructive meetings, phone call conversations and emails.

Despite changing employee demographics, Perkins employees remain strongly committed to the company. Many people work long hours, especially before deadlines, making sure that tasks get done properly and on time.

4.2.4 Six-sigma culture

In parallel to personnel changes, Perkins' culture has recently undergone a cultural adjustment due to the introduction of the six-sigma methodology. Six-sigma is a disciplined, data-driven approach to improving quality by eliminating defects (Eckes, 2001) mainly applied to manufacturing. It has been effectively deployed by many companies including General Electric, Motorola, Allied Signal and Caterpillar to reduce cost and waste, thus increasing profits. Six-sigma includes a suite of statistical analysis tools, the effective deployment of which requires well-defined, measurable data. This leads to difficulties when applying the technique to complex design process remodelling, because the resulting process improvements are often difficult to measure and cause-effect relationships are frequently obscure. Nonetheless, many employees also benefit from extensive six-sigma training, which results in an efficiency-conscious workforce, an attribute that is beneficial in its own right.

Thus, the design context at Perkins takes place against a six-sigma backdrop and is changing due to increased product and process risks and shifting employee demographics. The remainder of this chapter examines project planning in Perkins, focusing on how the company plans its design projects and what challenges arise in doing so.

4.3 Planning practice in Perkins

The case study set out to better understand project planning in industry. It aimed to find out what kind of plans are used, by whom and for what purposes. It also aimed to establish what kinds of problems arise due to the scale of industrial projects. These issues, along with other insights obtained during the case study, are discussed below.

4.3.1 Multiple types of plans

"People think a plan is a Gantt chart....I think that's a plan (pointing to a list of scheduled test-completion dates)" – Engine-test planner (ETP)

The most common plan-format used in Perkins is the Gantt chart. In addition to Gantt charts, however, several other information sources are used to plan and manage projects. These include Bill of Materials (BOM), Design Group Activity Schedules (DGAS), Microsoft Excel™ worksheets, Perkins Technical Operating Procedures (PTOPs), the New Product Introduction (NPI) process models and product confidence models (Fig. 4.2).

Although interviewees differed concerning what constituted a plan, all of the above aids are used by designers to organise and prioritise their work. Each different representation contained some information that is not captured by the others, although some information is duplicated in different plans. Also, the links between different representations are not always clear (Section 4.3.2). Determining how these plans fitted together is part of the planning puzzle.

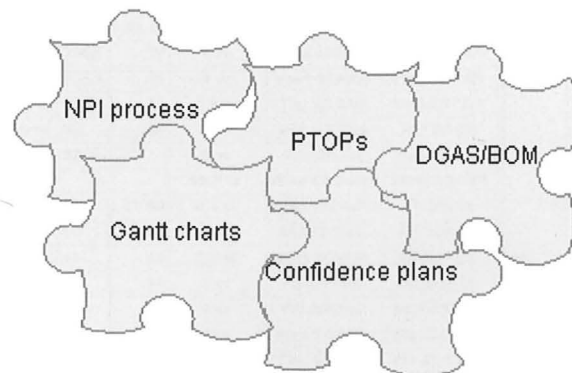


Fig. 4.2 Types of plans – different parts of the planning puzzle

Gantt charts

"Plans are too detailed...The project master-plan (a Gantt chart) currently has 34,000 lines!" – Senior manager (SM)

Gantt charts, generated using MS Project™, lie at the core of project planning at Perkins. Gantt charts show which tasks must be performed in which order, how long tasks are expected to take and which tasks are connected (Fig. 4.3). When a task-duration is changed, the MS Project™ software automatically updates the plan to show the impact on other tasks. The software can also be used to generate resource and cost profiles for a project.

Despite their usefulness, Gantt charts used in Perkins have many limitations: they fail to capture any information about uncertainty, such as uncertainty in task duration or task outcome; they do not provide an appropriate level of detail to support designers at a procedural level; they do not align well with the components listed on the Bill of Materials; the MS Project™ software is difficult to use causing at least one manager to create plans in MS Excel™, and Gantt charts provide no information about confidence in the product design. When using MS Project™, it is neither possible to represent the ramp-up of resources allocated to a task nor is it possible to show design iteration. Finally, the scale of plans was problematic – plans produced by the project office, in the form of MS Project™ Gantt charts, typically contained between 3000 and 5000 tasks while the project master-plan contained over 30,000 tasks. To overcome these limitations, several other plans were also used to complement the Gantt charts

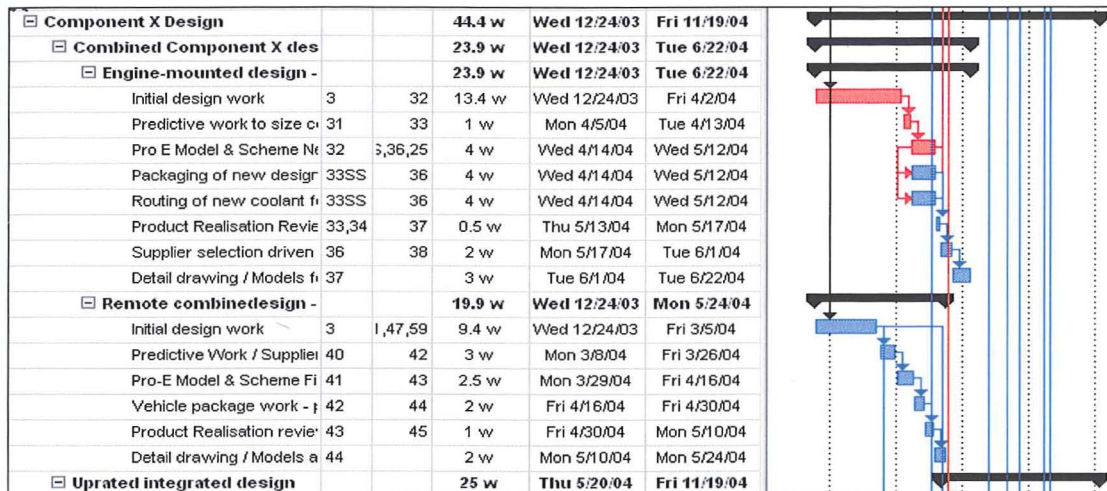


Fig. 4.3 Gantt chart excerpt (simplified for confidentiality)

"One of the platform team guys strung together a slightly different plan (in MS Excel™)... it's helping me to focus on how we could actually achieve the delivery of all the different sumps". – Team leader (LCT)

MS Excel™ was also used to create Gantt chart plans. These worksheets were created at an individual design-team level and are not used by all teams. Further, the scale and usage of MS Excel™ plans differ greatly between teams. In the testing facility, the approach is used extensively to optimise resource usage, while in the large component design group the MS Excel™ plans played a less significant role.

In both cases the plans were used to produce conventional Gantt charts but MS Excel™ was favoured over MS Project™ due to familiarity, ease of use and flexibility. The MS Excel™ package allowed the designers to manipulate project

plans in a manner that was not possible in MS Project™. No effort to achieve automated data exchange between the two packages was witnessed, although some duplication of information was evident. This was due to difficulties in interfacing between both software packages.

Process documents

"A process tells you the method by which you should do something and a plan puts that process into action saying when you're going to do it, how you're going to do it, what resources are required." – Project Manager PMP

Process documents, which are generic across different engine designs, are used to prescribe the order for groups of activities. PTOPs (Perkins technical operating procedures) are procedural documents which guide designers in performing tasks. They show task dependencies at a high level of detail and also capture local iteration between tasks. These are complemented by the generic New Product Introduction (NPI) process (Fig. 4.4), which decomposes the new product development into gateways. Both of these documents are highly linked to the Gantt charts; which define appropriate activities and milestones for each gateway in the process, while the former describes the specific work required to meet these milestones.

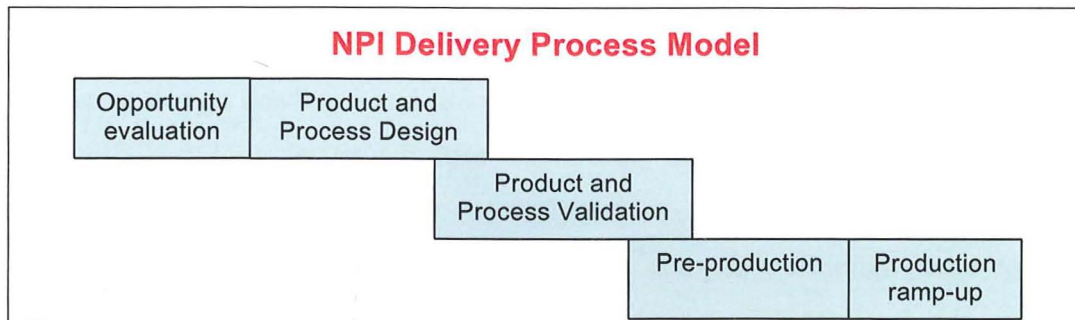


Fig. 4.4 Perkins NPI Process (simplified for confidentiality)

The PTOPs (Fig. 4.5) are also used to guide designers when performing work with which they are unfamiliar. Thus, they are especially relevant to novice designers. In contrast, the NPI document provides a detailed checklist of activities that need to be completed at different gateways, but provides little insight into how this work should be performed.

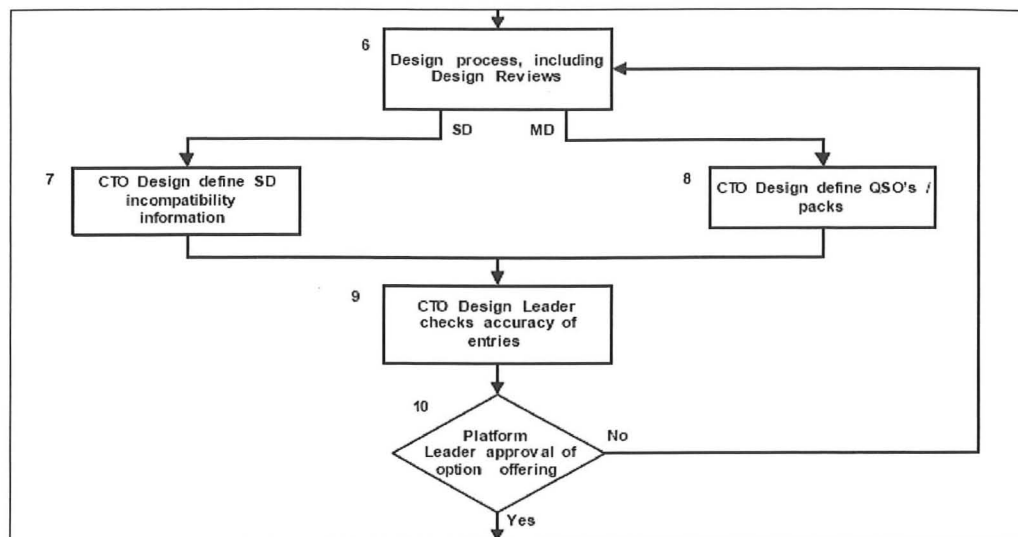


Fig. 4.5 Perkins Technical Operating Procedure excerpt

Bill of Materials

"The vision is that DGAS will drive resource and design activities" - Team Leader (LCT)

"I see DGAS as a knee-capping tool" - Process documentation manager (PDM2)

Sales engineers put together a product specification from customer requirements, which in turn was used to create the Bill of Materials for the product. Design team leaders combined this information with high-level project deadlines from conventional plans and lead-time durations based on previous projects to create Design Group Activity Schedules (DGAS) as shown in Figure 4.6. DGAS specifies component design deadlines and links them to the designer responsible. DGAS does not specify how long design activities would take, leaving the detailed sequencing of tasks in the hands of the designer.

DGAS has been widely implemented within the company, although some concerns are evident. When a designer adds his components to the DGAS list, he/she commits to a delivery date. Hence, some designers refused to add their components to the DGAS system until they were highly confident that they could achieve their commitments. As a result, some high-risk component-design activities were deliberately omitted from the DGAS schedule to avoid attention from management.

Item Number	Item Description Basic	Designer	Team Leader	Created	Design Release Required	Revised Drawing Release	Material Available	First Required Date
1757C0	CONNECTION	Simth J	Jones P	05-Feb-2004	16-Feb-2004	13-Feb-2004	15-Mar-2004	29-Mar-2004
1757C0	CONNECTION	Simth J	Jones P	05-Feb-2004	16-Feb-2004	13-Feb-2004	15-Mar-2004	29-Mar-2004
1757P0	BRACKET	Simth J	Jones P	04-Feb-2004	09-Feb-2004	13-Feb-2004	22-Mar-2004	05-Apr-2004
1757P0	BRACKET	Simth J	Jones P	04-Feb-2004	29-Mar-2004	06-Feb-2004	10-May-2004	24-May-2004
1757R0	L P FUEL PIPE ASSEM	Simth J	Jones P	04-Feb-2004	19-Jan-2004	06-Feb-2004	15-Mar-2004	29-Mar-2004
1757R0	L P FUEL PIPE ASSEM	Simth J	Jones P	04-Feb-2004	19-Jan-2004	06-Feb-2004	15-Mar-2004	29-Mar-2004
1757Y0	FUEL FILTER HEAD	Simth J	Jones P	05-Feb-2004	09-Feb-2004	13-Feb-2004	15-Mar-2004	29-Mar-2004

Fig. 4.6 DGAS schedule (simplified for confidentiality)

Confidence models

"Around here, for years, you hear people quoting a confidence level. We're 90% confident in that design. It's just a number – it does not mean anything." – Senior manager (SM)

The product-process link is a core topic of this thesis. At Perkins, design confidence data are used to connect both domains. In 2003, the company created a high-level mapping of the way in which different activities contributed to the confidence associated with the design of individual components. The confidence numbers were intended to provide an objective description of the maturity of different component-designs, in order to facilitate negotiation between different design teams. A more thorough discussion of the meaning of confidence, as used in Perkins, is given in Section 4.4.4.

Confidence data enables designers to identify low confidence components and determine suitable recovery plans if these components present a major risk to the project. Also, the confidence numbers provides an indication of which components are more susceptible to change at a given point in the process and which parts of the design are comparatively mature.

Table 4.1 shows abstractly how different activities contribute to design confidence within the case study company. Each activity is assigned a weighting. As the associated design work is carried out, the percentage completeness is estimated by design team leaders. This percentage is multiplied by the weighting to determine the confidence contributed by a given activity at a point in the process. The sum of the confidence contributed by different activities constitutes the total design confidence for a given component. The colour of the total confidence number indicates the component risk – green for low, orange for medium and red for high.

Component	Activity	Weight	Percentage Complete	Confidence Contributed
Crankshaft	Activity 1	50.0%	75.0%	38.0%
	Activity 2	10.0%	100.0%	10.0%
	Activity 3	10.0%	80.0%	8.0%
	Activity 4	30.0%	0.0%	0.0%
				56.0%

Table 4.1 Confidence model (simplified for confidentiality)

However, confidence information was not explicitly linked to the Gantt chart project-plans and was used more as a progress monitoring tool than a planning tool although it was used for both purposes. Also, the confidence approach was new to the company, having been conceived in 2003, and some assumptions about confidence were subjective, thus failing the original goal of objectivity.

4.3.2 Links between plans

"A number of people do work off their own agendas but their plans do not link to the overall project plans." – Finance and accounting manager (FAM)

"You've got an activity plan on one side and a resource plan on the other and they do not fit together – they do not even attempt to fit together" – Six-sigma black belt (SSB1)

Numerous difficulties arise in handling multiple types of plans simultaneously. Each of the different plans contains some useful information which contributes something to design project success. At the same time however, certain information is duplicated in different plans and it is difficult to ensure that they all contain consistent information. Further, a consistent nomenclature is not maintained across the different plans – the same tasks on Gantt charts and confidence models are given different names, an issue which can lead to confusion.

Broadly speaking, the links between the NPI process, the Gantt charts and the PTOPs are reasonably clear (Fig. 4.7). Deliverables at each gateway in the NPI process are translated into tasks on the Gantt charts. This approach is generally effective but leads to problems in planning intra-gateway tasks: some tasks must be completed quickly because they create information for downstream dependent tasks subject to the same

gateway. The NPI provide an insufficient level of granularity to guide planners in ordering these tasks.

Even when the Gantt charts have been produced they do not provide the designers with sufficient detail to carry out their tasks. Such information is contained in the PTOPs – they show, for example, exactly which steps must be performed when executing a specific design activity such as a component test. However, the PTOPs do not usually show how long the task is expected to take, information which is required to construct the Gantt charts.

Also, the link from Gantt charts to DGAS is usually clear, although the DGAS plans are simplified to component due dates and fail to show any information about the dependencies between different component designs. The dates for DGAS are lifted from the Gantt charts, although the process of updating DGAS in light of changes to these charts is not always seamless.

While most of the different plans are explicitly linked together, all links to confidence plans are extremely tenuous (Fig. 4.7). This may be due to the newness of the confidence models or the fact that they are focused on monitoring and control. Perkins was interested in linking Gantt charts to confidence plans. However, both plans use dissimilar terminology to describe the same tasks in some instances and frequently used different levels of abstraction (Gantt charts being more detailed). Also, linking of both plan formats presents software challenges – writing visual basic code to interface between MS Project™ and MS Excel™ is non-trivial. Development of a software interface would require standardisation of both plan formats, reducing their flexibility to future changes.

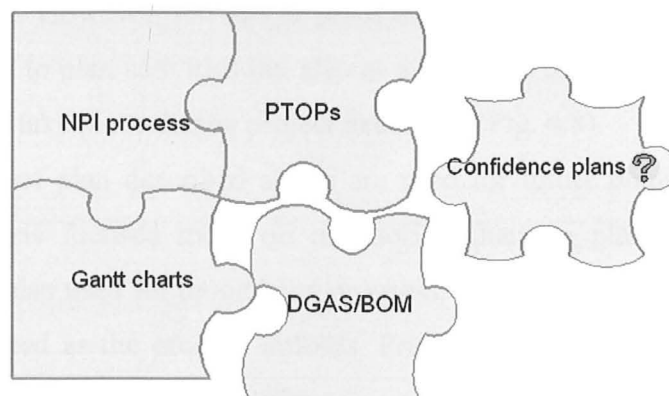


Fig. 4.7 Links between plans

Several problems arise when trying to use multiple plans concurrently: the lack of a single effective plan results in poor understanding of process connectivity and efforts are duplicated in maintaining several different documents. At the same time, even the sceptical team leaders concede that some plans, or parts of plans were accurate or that plans could be partially correct: for example: right in terms of task-order while providing poor estimates for task durations. They acknowledged that individual team-leader plans are often reasonably accurate, while delays are almost inevitable when trying to integrate activities across different teams.

When a process diverges from the planned route there is no easy means to update the plan or re-evaluate the project. When plans overlap, as is often the case, a change in one plan should be reflected by a change in another. However, dependent plans are not always simultaneously updated in practice, a factor which results in inconsistent plans and confusion for designers. Conversely, isolated plans result in poor design-overview and impede effective process trade-offs.

Although it can be advantageous in providing a simplified perspective on a complex problem, the concurrent use of multiple plans is problematic because efforts must be made to ensure that different plans are mutually consistent. Mechanisms for dealing with multiple plans are discussed in section 4.4.5.

4.3.3 Uses of plans

"There's a problem with planning here [in Perkins]. It needs to be much more about proactive planning rather than reactive reporting." – Six-sigma black belt (SSB1)

By definition, the act of planning is forward-looking; it concerns something that has not yet happened. However, the use of plans in industry is more complicated – they are not only used to plan activities but also as a means to track progress and to record the events which take place during project execution (Fig. 4.8).

All of the types of plan described above are used for future planning, although the confidence models focused more on monitoring than on planning. NPI gateway deliverables are also used for monitoring progress, but neither the NPI process nor the PTOPs are updated as the process unfolds. Process records are instead captured by Gantt charts, DGAS and confidence plans.

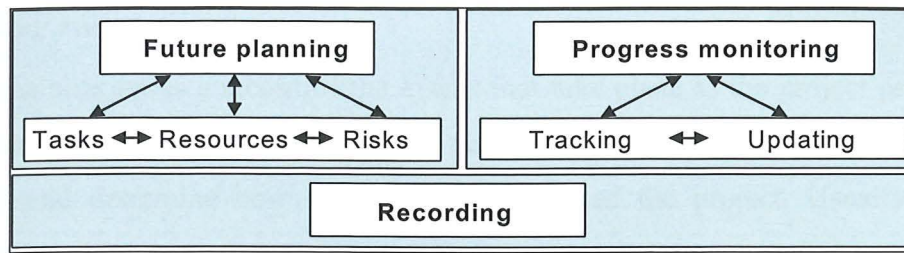


Fig. 4.8 Chronological uses of plans: future planning, monitoring and recording

Future planning

Chronologically, the first use of plans during a design project is to define its scope, i.e. to provide an estimate for a project's feasibility in terms of the likely cost and duration. In order to scope a project, an initial plan is established based on the NPI process document and from previous project plans. The project duration is estimated by summing task-duration estimates and also based on predicted design deadlines. In the case of Perkins, these deadlines may be determined by emissions legislation.

The initial project Gantt chart can be quite abstract: it need not concern the detailed work breakdown at a designer level. However, this detail is exactly the information required by team-leaders – once the project is approved, such middle managers must decide how to allocate specific tasks to their team members. This requires the addition of greater detail to the plans (the issue of appropriate detail in plans is discussed in section 4.4.2). Also, such detailing of the plan is an ongoing activity: tasks which will take place in the near term are planned to a high level of resolution, while downstream tasks are often planned abstractly early in the design process. Detail is added later as more information becomes available.

Monitoring and updating

As the project progresses, Gantt charts are used to monitor progress against the planned targets. This helps designers and managers determine when they are on time and when they are falling behind schedule. The Gantt charts also provide some information about how tasks are linked together and how delays are likely to impact others, although they rarely provide a complete picture. In this sense, plans are used to control the project. Milestones push designers to meet certain targets by predefined deadlines. The plan can be used to establish metrics against which to judge progress at a given point, thus providing an indication of the likelihood of meeting the overall project requirements. Monitoring and updating also allows management to predict downstream problems which are not foreseen at the project outset.

Recording events

The plans also act as a record of the events that take place as the project progresses. They are sometimes used during project post-mortems to establish which tasks were delayed and determine how these delays influenced the project. Usually, several different versions of a plan, saved at different stages during the project are available upon completion of the project. Management and planning personnel can compare and contrast different versions of the same plan to identify trends and to provide insights into when plans accurately predicted the project execution, when the actual events diverged significantly from those planned and how plans evolved over time. Such information can be used to guide the planning of subsequent projects.

At Perkins, however, in-depth project post-mortems are not always carried out, because people become engulfed in subsequent projects, although some form of post-mortem is recognised as good practice.

Other uses of plans

"The financial plans have driven what engineering can and can't do. It's probably the wrong way around. An engineering plan should plan what the finances should be. When you get into this situation of financial plans driving engineering, you suddenly fall into the trap of shorttermism." – Finance and accounting manager (FAM)

As well as the temporal distinction in the uses of plans, different users utilise plans to realise a diverse range of objectives (Fig. 4.9). Engineers use plans to co-ordinate their activities; team-leaders use plans to manage their resources against predefined deadlines; high-level managers use plans firstly to predict the feasibility of new projects and later control projects against predefined targets; finance use plans to allocate budgets on a monthly basis; sales and marketing people use plans to determine when interim or final products can be shipped to customers; and external consultants use plans with the goal of improving the design process. Nearly all of these activities are performed on the same plan – the Gantt chart – although different levels of abstraction, or even different content, would be appropriate for the different applications. As a result, plans tend to become large and contain some information which is not relevant to the purpose at hand. The penknife analogy is appropriate: while a single plan performs several functions, it is not the most effective format for any of these functions.

Recording events

The plans also act as a record of the events that take place as the project progresses. They are sometimes used during project post-mortems to establish which tasks were delayed and determine how these delays influenced the project. Usually, several different versions of a plan, saved at different stages during the project are available upon completion of the project. Management and planning personnel can compare and contrast different versions of the same plan to identify trends and to provide insights into when plans accurately predicted the project execution, when the actual events diverged significantly from those planned and how plans evolved over time. Such information can be used to guide the planning of subsequent projects.

At Perkins, however, in-depth project post-mortems are not always carried out, because people become engulfed in subsequent projects, although some form of post-mortem is recognised as good practice.

Other uses of plans

"The financial plans have driven what engineering can and can't do. It's probably the wrong way around. An engineering plan should plan what the finances should be. When you get into this situation of financial plans driving engineering, you suddenly fall into the trap of shorttermism." – Finance and accounting manager (FAM)

As well as the temporal distinction in the uses of plans, different users utilise plans to realise a diverse range of objectives (Fig. 4.9). Engineers use plans to co-ordinate their activities; team-leaders use plans to manage their resources against predefined deadlines; high-level managers use plans firstly to predict the feasibility of new projects and later control projects against predefined targets; finance use plans to allocate budgets on a monthly basis; sales and marketing people use plans to determine when interim or final products can be shipped to customers; and external consultants use plans with the goal of improving the design process. Nearly all of these activities are performed on the same plan – the Gantt chart – although different levels of abstraction, or even different content, would be appropriate for the different applications. As a result, plans tend to become large and contain some information which is not relevant to the purpose at hand. The penknife analogy is appropriate: while a single plan performs several functions, it is not the most effective format for any of these functions.

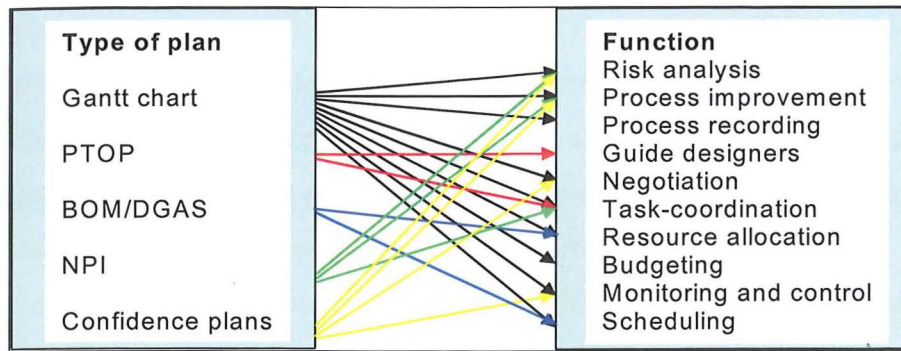


Fig. 4.9 Roles of plans in Perkins

4.3.4 Creating and owning plans

"In CAT they have a very different attitude to planning. The project office are effectively assistant project managers and they are far more empowered. They have the confidence and background to be able to ask far more questions whereas here it's basically an entry level position." – Six-sigma black belt (SSB1)

Project planning is the combined responsibility of three different groups within Perkins: design engineers, managers and plan administrators all contribute. In the case of Gantt charts, the plan administrators work with the engineering managers to obtain estimates for task durations and dependencies. Some experienced designers may also be asked for input at this stage. Some dependencies can also be identified based on the process documents (both NPI and PTOps).

Few plans are completely new; in contrast, plans from previous projects often constitute a useful starting point by providing partial plans. The degree of novelty depends on the technical innovation associated with the design and the time constraints involved. Some chunks of the design process, for example testing schedules, are repeated from one design to the next. In contrast, the tasks associated with novel design content cannot be copied from a previous plan. Severe time constraints may require changes to the partial plans even if the actual design content is similar – for example, it may be necessary to increase the degree of concurrency.

Planning constraints can also arise due to legislation, customers and suppliers: legislation determines a latest possible date for new product introduction; customers will often request engines before this point, as new engines must be integrated into vehicles or other products, and supplier lead-times dictate intermediate deadlines within the design process. All of these constraints help define a structure for the plan.

In Perkins, team leaders within the Core Technical Organisation (CTO) work with the plan administrators to create team plans, in the form of Gantt charts. Links between team plans are captured in the project master-plan, a Gantt chart plan which resembles the NPI process document but provides much more detail about task durations.

In contrast to Gantt charts, which are mainly produced by team-leaders and plan administrators, Design Group Activity Schedules (DGAS) are created and administrated by individual designers (Table 4.2, Fig. 4.10). Necessary data to construct these plans is taken from the Bill of Materials (BOM). Confidence-based plans and MS Excel™ sheets are created by the team leaders, although they may be updated by designers. The plan administrators are not involved in the creation or utilisation of confidence plans. Perkins Technical Operating Procedures are owned and updated by a small group of engineers but used widely throughout the company, and the NPI process is based on advice from external management consultants.

Type of plan	Creator	User
Gantt chart	Project office, team leaders	Designers, team leaders, accounting, finance
PTOP	Designers	Designers
BOM/DGAS	Designers	Designers, purchasing
NPI	External consultants	Project office, Design teams
Confidence plan	Team leaders	Designers, team leaders

Table 4.2 Creators and users of plans

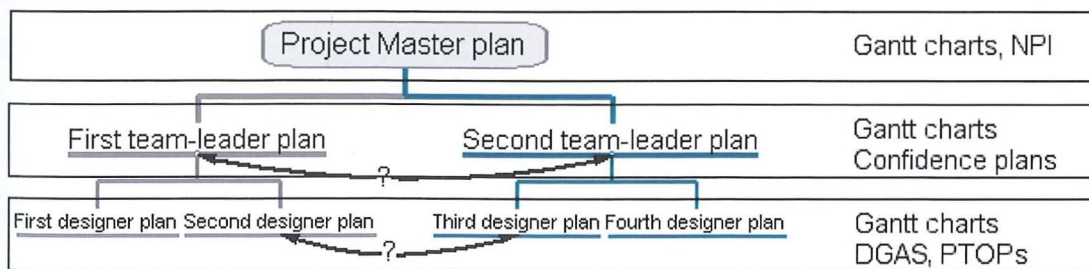


Fig. 4.10 The hierarchy of plans

4.4 Project planning challenges in Perkins

The focus of this chapter lies on the planning of design projects with a view towards coordinating design activities and minimising risks. In doing so, planners and managers encounter numerous challenges – from modelling and representing uncertain information and contingencies subject to varying degrees of scale and connectivity level, through planning for rework and iteration, to capturing the inherent dependencies between product and process. These challenges are discussed below.

4.4.1 Modelling uncertainty and contingency

“There is a general lack of decision on whether we have contingency included or it’s not included – it’s a massive thing. This came up time and time again. If we declare contingency, then it’s squeezed; if we do not declare it, then we assume it’s in there anyway.” – Six-sigma black-belt (SSBI)

“I’m not aware of them (contingencies). People plan for success and do not build any contingencies into their plans.” – Finance and accounting manager (FAM)

Because plans are created at a point when the actual process has not yet taken place, they are subject to numerous sources of information uncertainty. Such uncertainty makes it difficult to predict what will happen as the project unfolds. Often, uncertainty is hidden within the tasks: e.g. tasks contain contingency for delays and low-quality task outcomes. Ambiguity in task definitions can lead to the impression that a plan is being followed closely, while the actual events differ significantly from those envisaged by the planners. This approach, however, leads to problems in modelling and analysing processes and when determining whether the project is really on track for success.

There are a large number of uncertainties when creating project plans. At Perkins, some concern the technical challenges that the company will face in designing the new product. Others reflect changes to the process in order to meet emissions deadlines. In addition, task durations and outcomes vary according to experience but the design team itself may be unknown during the early planning phase. Similarly, the strength of dependencies between tasks varies – some previously sequential tasks can be performed in parallel but at increased risk to the project. As a result, it is extremely difficult to create accurate plans.

During the Perkins case study it became clear that designers, and especially design managers, spend a large part of their working hours performing important, but frequently unplanned and unplannable work (answering emails, phone-calls, attending meetings, dealing with unforeseen problems). Issues arise on the shop-floor which require immediate attention; co-workers fall ill and work is redistributed; boardroom decisions are made to change supplier resulting in slight component variations which drive major levels of unplanned re-design work and hours are spent in meetings, answering emails and taking phone calls. Uncertainty abounds in the design process – the challenge is to create plans which remain useful in this context.

Contingency

“Just from historical understanding and knowledge of working on other projects you build in time that can cope with the amount of churn that can happen...ideally you plan so there’s some sort of leeway.” – Team leader (LCT)

One mechanism for dealing with uncertainty is the use of contingency, but this is not a simple approach in practice. When managers know which tasks are likely to prove problematic, they can introduce contingency into their plans to enable them to deal with these problems. Predicting problematic tasks, however, is not always straightforward.

One manager (LCT) commented that his designers always give optimistic estimates for their task durations and fail to account for things going wrong. Another (PMP) noted that some designers give optimistic estimates for task-durations while others include large contingencies. He adjusts the level of contingency differently on an individual basis for members of his design team. Yet another manager (CaeT) commented that the amount of contingency required depended on the experience of the person performing the task. This also introduces a problem when planning contingency factors, as it is not always clear who will perform tasks planned for the medium-term future. Further, contrasting attitudes to contingency in plans were also noted: designers thought that managers did not want it, but managers thought it should be included.

Challenges also arise in representing, communicating and negotiating contingencies. None of the plans used in Perkins explicitly showed contingency, and it is impossible to distinguish tasks which included contingency from other tasks by examining plans. Under such circumstances, it is likely that contingency for the same event was

sometimes included more than once, particularly when planning at different levels of abstraction, thus introducing process inefficiencies. At Perkins, designers, design team-leaders and plan administrators all play a role in estimating task durations. It is possible that contingency for some risks is duplicated in the plan, although finding such duplication is extremely difficult because nobody knows about it. When contingency is not explicitly captured by plans, different team leaders cannot easily determine whether their colleagues are likely to cause delays, a factor which encourages them to hide potential delays to their own components (Ford and Sterman, 2003).

In addition to time-based contingency, Perkins also exploited solution-based contingency. For example, if a design was not completed on schedule, small volume production would begin with soft-tooling. While this approach was not ideal due to increased manufacturing costs, it was often preferred in the Perkins design context of tight emissions deadlines.

Miracle boxes

"We do not know how it's going to be done. We know that something is going to happen here so we won't think about that. We'll call that six-sigma and a project will be done on that. That's a miracle box". – Six-sigma black belt (SSB1)

A second mechanism for dealing with uncertainty at Perkins is the use of abstract planning in the form of miracle boxes. These are a representation of high-risk tasks on Gantt charts, sometimes highlighted in a distinct colour (Fig. 4.11). Usually the miracle boxes refer to tasks, or groups of tasks, which must be performed significantly faster than on previous designs, without reducing the design quality (a "miracle"). Often these tasks become the focus of six-sigma projects, which means that they can draw on extensive knowledge and resources within the company to develop suitable solutions.

Miracle boxes can also be thought of as black boxes with the function of risk localisation. Their output states are defined, allowing the planning of dependent downstream tasks, but it is unclear what happens within the miracle boxes. Thus the activities within the miracle boxes are initially planned to a low level of detail. Later, as more information becomes available, the input state becomes better defined and detail is added to the miracle boxes. To some extent, this allows dynamic planning of the high-risk activities in reaction to highly-specific design-context information.

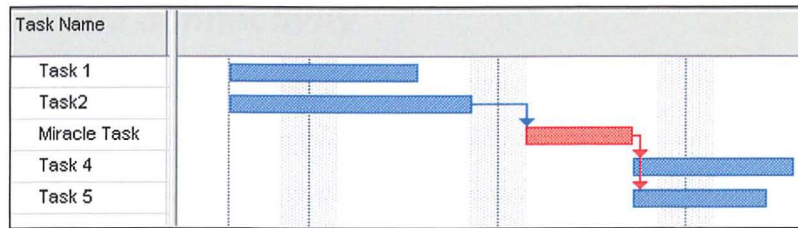


Fig. 4.11 A miracle box

Representing uncertainty

Despite the fact that much of the information upon which project plans are based is uncertain, many plans (Gantt charts, PTOPs, NPI, DGAS) display information in a binary manner and hence portray little information about project risk. Gantt charts, for example, do not contain any information on uncertainty, do not account for ambiguity in task definition and do not model rework. As a result, they fail to capture the risks associated with the project and can lead to a false sense of optimism. This is especially true when the plan involves a number of uncertain events as is often the case with design. In such cases, cumulative uncertainties are almost sure to result in a process execution which differs from the plan. Designers and managers know that previous plans have been inaccurate. As a result they are wary of plans that fail to account for uncertainty.

Deadline driven design

"There's two ways of doing [planning] it. I either work forward from the concept design and say, right, this guy's available, it's going to take him so long to do a design, so long to procure it... From the other end, the customer requires it by this date. We know it's going to take 26 weeks to procure so they need to have the design available to procure 26 weeks before." – Team leader (LCT)

The output states of miracle boxes are often defined based on project deadlines and supplier lead-times. Perkins uses a combination of forward and backward planning or deadline-driven design to achieve its goals: forward planning looks at how long each task will take and sums up the total duration accounting for dependencies, while backward planning looks at the deadline and plans backwards from this point. As both the initial starting point and the later test schedules and completion deadlines are usually well defined, the challenge arises in planning the activities which take place roughly half-way through the design process as the design context at this point is very difficult to predict. This is where miracle boxes are most frequently deployed. Reasons for difficulty in planning the intermediate stages of projects include problems due to scale and iteration, as discussed in the following sections.

4.4.2 Scale and connectivity

"We spent so much time creating plans with 2000, 4000, 17,000 lines and that does not tell you anything." – Six-sigma black belt (SSB1)

"We're so busy beaver away doing all this detail; we very rarely pick up all these dependencies". – Team leader (LCT)

Some plans at Perkins are reasonably small involving few tasks, which are relatively independent from the rest of the design. These plans rarely lead to problems. Large project plans, in contrast, represent tens of thousands tasks – this introduces planning challenges relating to scale and connectivity. At Perkins, the links between different team-leader plans are often unclear or unknown, a situation which is aggravated by a lack of technical knowledge in the project office team: nobody in the project office – the group responsible for plan integration – had an engineering background. A number of other problems, such as understanding the implications of task-connections, were also noted. Much of the academic literature focuses on small, simple models. This section, in contrast, considers the challenges of large-scale project planning.

Achieving an appropriate level of detail

It is important to distinguish between the level of detail in plans and the level of detail in models (Fig. 4.12). A model is an abstraction of the process; depending on the degree of abstraction, the scale of the model will vary even though the scale of the process is unchanged. However, as models grow more abstract, they include increasingly less detail, a factor which can detract from their utility. At the same time, however, overly detailed models can prove difficult to represent and analyse. At the same time, it is not easy to decide what to include or omit in an abstracted model.

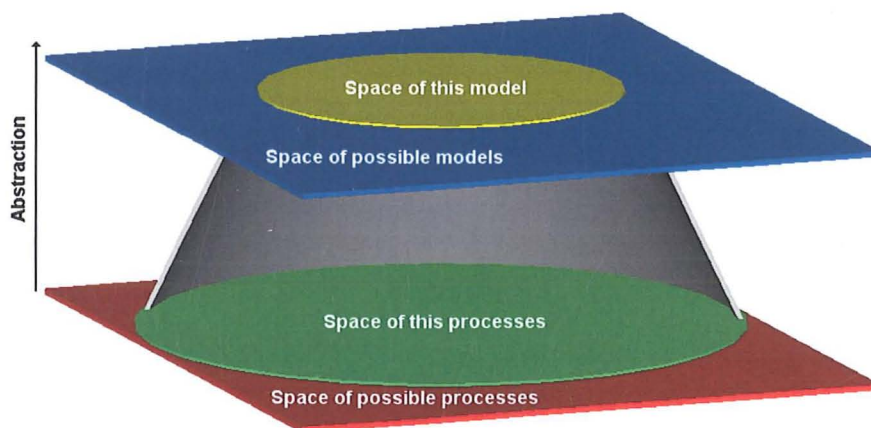


Fig. 4.12 Abstraction of the process to the process-model

Different users of plans desire different levels of detail. A team leader (LCT) complained that the timing information for tasks was artificially precise and sometimes unrealistic. Another interviewee, a process manager (PDM2), stated that:

'If it says on the plan that an engineer should be doing this activity in the morning and another in the afternoon and he really gets distracted from the original [task] but gets some of next week's activity done, then there isn't a problem in meeting the project deadlines although he's way outside the plan.'

A finance manager, whose job it was to allocate budgets for the individuals made a similar comment, stating that the level of detail in the plans made them useless for his work in finance and that some method of producing more abstract plans was required. While designers want guidance at an activity level, finance managers are only interested in a more abstract level, such that they can plan budgets for different teams. From their viewpoint, the plans are far too detailed and the high quantity of information obscures the important content.

Getting the right information into plans

"If those who were reluctant to plan in detail are forced to do so, they might deliberately ensure that the plan fails so as to avoid the tedious planning activity in future" – Senior manager (SM)

A comparison between the comments made in interviews and the data contained in project plans illustrated that many information driven task dependencies, which are well known to designers and defined in process documents, are not explicitly captured in the project plans. The average number of connections per task on most Gantt charts is less than one, while the average number of connections in most published literature is closer to six. Although task timings on project plans are driven by task precedences, the absence of explicit links means that updating plans becomes difficult.

Some team leaders make a thorough effort to plan well, while others use plans as little as possible. The sceptics argue that they were not convinced of the value of planning. They can produce examples of previous projects which had diverged significantly from plans and, as a result, are sceptical about the latest planning initiatives.

Understanding the importance of task-dependencies

"Dependencies are not considered sufficiently. I think plans work within functions [functional teams] but where we have problems is at interfaces. It comes back to dependencies." – Senior manager (SM)

The scale of the Perkins project plans makes it difficult to understand connections between tasks. Links between tasks are extremely hard to capture; even when they can be captured, their visualisation poses challenges. Hence, it is difficult to understand how tasks are linked together by looking at project plans. Both planners and managers are overwhelmed by the volume of information available and have difficulties in determining what is important.

Some components are highly connected to other parts of the design, while others are reasonably isolated. The design of more connected components often proves more difficult because they are more likely to be affected by changes elsewhere. Although designers and managers are highly aware of which components are likely to change, this knowledge is not reflected in the plans, and there is little understanding of the implications of product connectivity for planning.

4.4.3 Iteration and rework

"If you get it wrong and have to do an iteration, it affects the timescales quite badly. There could be a high risk factor of one of those things going wrong and then you end up doing rework – this guy here did not get the right information in time and suddenly this one here's going to overspend as well." – Project Manager (PMP)

Literature on process modelling highlights iteration as a key issue in design (Chapter 3). Perkins designers and managers recognise rework as an important source of difficulty in planning and note the existence of a trade-off between performing rework and reshuffling of downstream tasks. At Perkins, some rework is covered by contingency in plans – an approach which only works when designers have a good understanding of the expected rework and can use conservative task duration estimates to account for it. However, ever-tighter deadlines make it increasingly difficult to include such contingency.

Different types of iteration in Perkins align closely with categories reported in the literature and include iteration to converge on a solution, re-occurring tasks and rework due to undesired task-outcomes (see Section 3.2.2). However, the Perkins

study also brought to light the trade-off between rework and reshuffling of dependent downstream tasks.

During a discussion with a senior manager (SM) at Perkins, it became clear that the response to a task-failure is not always rework. Conversely, the project may proceed subject to the reshuffling of downstream tasks – even a failed test may provide some useful information and hence may not need to be repeated: modification to another test may be sufficient to verify that the redesigned component meets its requirements, especially in the case of minor changes. Even when rework is performed, downstream tasks may continue subject to a lower information input state, as delaying them may not be a realistic option – the downstream task order may be reshuffled in response to the upstream failure, in order to minimise its impact on the process.

In addition, designers sometimes obscure the effort associated with reworked tasks: if some tasks go well and are completed ahead of schedule while others fall behind, they may shift their efforts to the lapsed tasks without updating the plan. Where designers are required to sign off work against a budget, they may choose budgets that are already approved but which do not reflect the work being done. Thus, planners cannot always easily determine how the designer's time and budget are actually spent during project execution, even when rich information on previous projects is available. This makes accurate planning and effective utilisation of plans difficult; rework identification, measurement and analysis is particularly challenging.

4.4.4 Confidence-driven planning

“(Confidence) is a mixture of programme confidence and product confidence....it's partly objective and partly subjective...it's not clear what people mean by confidence.” – Senior Manager (SM)

As noted in Section 4.3.2, the link between task-based plans and the product confidence model was almost non-existent. As a result, the company had no way of predicting how changes to the tasks were likely to affect the likelihood of achieving the desired product confidence. Optimisation of the product and the process was performed independently and the effects of changes in one domain propagating to the other are not considered. Hence, changes to the design process could have a negative impact on product quality. For example, the introduction of new software, which aims to expedite the design process, can result in design errors which detract from product quality.

Another example, which illustrates the importance of the product-process link, is rework (previous section). Failure to design product features to the desired level of quality requires rework in the process. A means to predict the downstream effect of rushing early tasks, and hence failing to complete them to the desired level of rigour, is lacking at Perkins, a problem which the company readily acknowledges. A link between both task-based and confidence plans would be beneficial to both designers and managers in helping them prioritise different tasks.

To address these challenges, Perkins is moving towards confidence-driven design, an initiative that will require considerable changes to the way in which projects are planned. The potential benefits are manifold: having a consistent set of activities in both the plans and the confidence models would be useful for planning, controlling, managing and reviewing projects. In light of the effort involved in integrating the confidence data with the Gantt chart plans, however (Section 4.3.2), thorough consideration of the merits and limitations of the approach is necessary.

Defining confidence

A clear definition of confidence does not exist in Perkins. Currently the word is loosely defined to capture several different meanings:

- *completeness of design in terms of activities performed;*
- *degree of solution refinement / solution maturity;*
- *predicted mechanical integrity;*
- *design quality at a point in time;*
- *confidence that the design will meet the specification.*

Further, confidence is acquired based on the 'gut-feelings' of experienced designers and also from evidence from tests. There is likely to be a gap between gut-feeling and evidential confidence (objective confidence based on tests), as evidence is only gathered through analysis and testing, while designers can claim confidence during earlier activities such as drawing. The fact that confidence can mean different things is problematic.

Despite these criticisms however, many team-leaders felt that the notion of confidence was useful and took the pragmatic view that even an ill-defined notion of confidence is better than none at all. Also, many had a good understanding of what others meant when they quoted different confidence numbers.

Assigning confidence across different design phases

The design process at Perkins is broken up into staged gateways. Perkins was trying to figure out whether confidence should reach 100% in each phase or only reach 100% by the end of the project (if at all). Originally, confidence data was only used for the testing phase, but the approach had been expanded to cover all phases of the design process. When carrying over confidence between gateways, a weighting needs to be attributed to each phase – this weighting differs by component. The potential existence of undiscovered rework introduces further complications. As a result of these issues, Perkins is concerned about the validity of the current multi-gateway confidence numbers.

Deciding the timing of confidence contributions

Deciding on the timing of confidence contributions is also challenging. Confidence can be added as a task progresses or at the end of the task. Some engine-testing tasks have a long duration. In reality, much confidence is gained early in the test, if the component does not fail. Further confidence is gained for each hour of testing, and the final confidence boost is realised during the engine-strip. However, breaking up the confidence for a single task into multiple stages introduces extra work for designers, and it is not clear whether the resulting benefits justify the required effort.

Allowing for new content / experience

Most design confidence comes directly from performing tasks. Some confidence, however, can be assumed based on experience from previous designs. Senior designers decide when confidence can be carried over from a previous engine programme, from the supplier's experience and/or based on competitor analysis. Broadly speaking, there is agreement in principle between team-leaders that experience should contribute to design confidence, but practical difficulties remain in deciding the appropriate weighting for different components.

Granularity

Although confidence is currently only applied to components at Perkins, it could be applied at several different levels of granularity including 1) whole engine level; 2) system level and 3) feature-level. It may also be useful to track confidence associated with performance characteristics such as fuel consumption, heat-dissipation capacity, emissions compliance, and noise, vibration and harshness (NVH) performance. While this is theoretically possible, and likely to be useful, it requires appropriate weighting

of different components within systems, or whole engines. Deciding on these weightings is likely to prove difficult in practice.

Resistance to integrating confidence plans and Gantt charts

Although the company was already using confidence to track design-progress, there were no projected confidence targets based on the project plans. Team leaders were concerned that linking both models would result in such targets which would then be used as an undesired performance metric. They were also worried that linking both plans would create significant work and that the rewards would not justify the effort required. This concern was overcome by writing software code to interface between MS Project™ and MS Excel™ and demonstrating the results on a simplified plan. Currently, both types of plan are updated separately and the team leaders could see the benefit of automatically updating the confidence plans.

In summary then, the use of confidence plans to link product and process data is a theoretically useful concept but numerous practical barriers impede its adoption in industry.

4.4.5 Coping with imperfect plans: the importance of overview

Despite the challenges of planning and managing complex projects, Perkins succeeds in delivering quality products on time and within budget. This section explores different factors which help Perkins achieve project success.

Overview of the product, the process and the organisation

It was determined from the study that both tacit overview and experience play a significant role in overcoming the limitations of plans: overview allows designers to see how their decisions affect others within the organisation, while experience helps them avoid mistakes. Although both overview and experience are closely related to expertise, they differ in many respects. While much expertise is focused on the depth of knowledge which individuals possess, overview is more concerned with the wide breadth of understanding acquired over time (Flanagan et al., 2006).

Product overview

"You can't just put blinkers on and work on your own component – how it interacts with other parts of the engine must be considered" – Mechatronics manager (MeM)

Senior designers observed during the case-study have an excellent overview of their product, knowing in detail how different components are connected and which

components are likely to change as the design progresses (Fig. 4.13). Although this overview was sometimes biased by the designer's experience – designers who have worked extensively on specific components are likely to have a deeper understanding of these components than of other parts of the design – it nonetheless allows them to consider several diverse factors at once and hence assess trade-offs. They understand the needs of their colleagues without requiring explicit information and thus avoid unnecessary interaction. Although their work is often informal and based on tacit knowledge, its importance cannot be overstated (Flanagan et al., 2006).

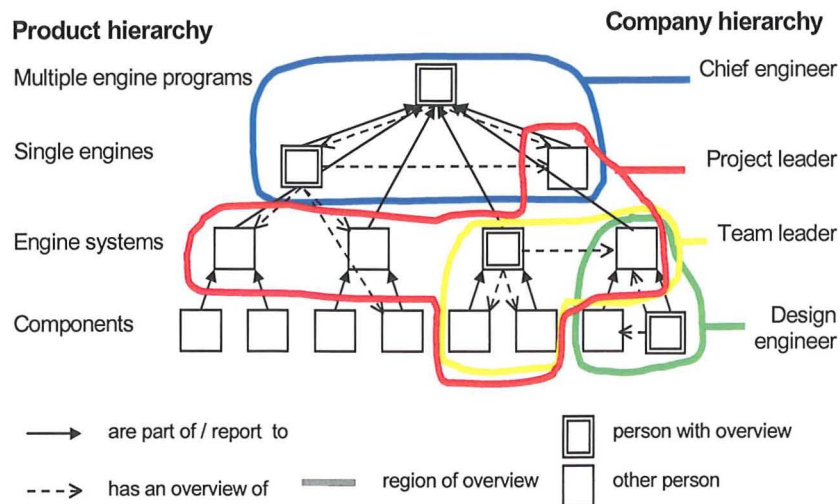


Fig. 4.13 Product and organisational overview in Perkins (Flanagan et al., 2006).

Organisational overview

"The org (organisation) chart does not tell you what they know, how long they've been doing the job" – Engineering Designer (ED)

Management overview of people also plays an important role by putting the right teams together to tackle the most pertinent problems. Using overview, they understand how different teams interact and where problems are likely to take place. At a designer level, overview of the organisation is critical in fostering an atmosphere where communication between designers can flourish. Designers often use informal channels to obtain the information they need so as to address specific problems. Even when the information is documented, brief conversations to address very specific queries can be a much more efficient means of obtaining information. Further, people who have previously worked on similar issues often include useful anecdotes on relevant experiences that are not captured in project reviews. A good overview of the organisation is valuable in knowing whom to contact and which questions to ask.

Process overview

"I certainly spend quite a bit of time stepping back from what the person needs to know and trying to explain to them where they fit in the big picture" – Mechatronics manager (MeM)

In addition to having a thorough product knowledge, experienced designers and managers have a good overview of the design process and are able to overcome the limitations of individual plans by using a tacit understanding of the relationships between different plans and between different tasks within the same plan. Also, overview and experience play a key role in progress monitoring and dynamic project management, and thus are central to project success.

Experienced designers are aware of the shortcomings of the plans they use and offset these limitations using various strategies. They can accurately predict how long a task will take and have the technical skills to compensate for unexpected problems that could lead to delays. They keep multiple types of plans in mind and resolve conflicts between them in an ad-hoc manner. Tasks are also prioritised based on overview – they know which tasks are most important in the context of the overall project constantly reshuffling activities to ensure that everything is done on time. By seeing the bigger picture they avoid the trap of local optimisation of their tasks at the detriment of the entire project.

Overview of other parts of the design is highly valuable in dealing with emergent requirements, managing rework and reshuffling plans. Experienced team-leaders at Perkins have a good understanding of which components are likely to fall victim to emergent requirements and how these changes will propagate to components within their teams. When possible, they use this knowledge proactively to schedule design activities with the goal of reducing the level of potential rework.

When the project falls behind schedule, experienced engineers are more likely to know 'the right way of doing things the wrong way' and they have the combination of experience, expertise and overview to overcome the limitations of the project plans. Tasks are also prioritised based on overview – they know which tasks are most important in the context of the overall project and constantly reshuffle activities to ensure that all tasks are completed on time.

While overview has many advantages, it can also result in problems. Experienced designers are acutely aware that a change to the project plan can cause problems and

can become resistant to change as a result. Because they have such a broad understanding of the design, they are difficult to argue against once they become set in their ways. Hence, overview can act as a barrier to the introduction of new ideas within a company. Nonetheless, it is generally a positive and necessary force in realising successful projects.

Other factors in project success

The dedication of Perkins staff is another important factor in allowing projects to succeed despite the limitations of plans. In many cases, deadlines produced early in the design process based on imprecise, insufficient and incorrect information assume a life of their own as the project progresses, and employees go to great lengths to meet these deadlines. Thus, even when the original task-duration estimates are poor, people manage to stick to the plan, sometimes by working overtime and/or weekends.

Other factors which help projects succeed are the maturity of diesel-engine technology, the effectiveness of the NPI and PTOP documents and the successful execution of six-sigma projects (Section 4.2).

4.5 Comparison with other design case studies

Several case studies, performed by other researchers, are examined here with a view towards better understanding the generality of the above observations. While these studies were not focused on planning, they nonetheless provide insights into industrial practice, which are relevant to this research.

4.5.1 Observations from EDC case studies

Case studies performed within the EDC (Fig. 4.14) are presented separately from external research because 1) interview tapes and transcriptions for most of these interviews were available for analysis, and 2) it was possible to discuss the interview findings at length with the academics that performed the studies.

Westland Helicopters

Studies at Westland Helicopters by different members of the EDC led to the conception of the Signposting modelling framework (Hamilton, 1999) and the engineering Change Prediction Method (Simons, 2000). The latter method is based on the aggregate likelihood and impact values for changes propagating between different components. The Westland's study confirmed the findings of the Perkins study in

several respects: as in Perkins, the design process at Westlands was riddled with sources of uncertainty, and challenges arose in building appropriate models of the design process. Unlike Perkins however, nobody had a complete overview over the entire product (Eckert et al., 2004b). Similar observations were made in other studies as discussed below.

Perkins Engines study on engineering change

Jarratt (2004) performed an extensive case study into engineering change management at Perkins Engines. Although his work was focused on predicting change propagation, it highlighted several issues which are also relevant for project planning. In fact, distributing the workload associated with engineering changes is, in itself, an interesting planning challenge. Jarratt's work also highlighted the importance of designer overview in realising successful products, and showed that managing engineering changes was a problem for industrial practitioners.

Automotive Consultancy

A study by Eckert and O'Donovan highlighted planning challenges in an engineering consultancy (Eckert and Clarkson, 2003; O'Donovan, 2004). In some cases the plans were produced retrospectively to document completed work, rather than proactively to increase process efficiency. In the absence of official plans, it was noted that the company used several other documents in parallel to realise successful designs (including Gantt charts, quality plans and Bills of Materials). Unlike Perkins, where the work atmosphere was very positive, personal animosities were observed at the consultancy, which tended to aggravate planning challenges.

Rolls Royce Aero Engines

Several studies have also been performed at Rolls Royce Aero Engines by O'Donovan (2004), Jarratt (2004) and Wynn et al. (2006). These studies have shown that challenges arise in planning Aero Engine design which are not evident when planning diesel engine projects – planning iteration is recognised as a major challenge at Rolls Royce but is less of a concern at Perkins. This may be due to the differing levels of complexity associated with both products. Other problems, such as modelling and representation challenges and difficulties associated with planning subject to uncertainty, were equally relevant to Rolls Royce and Perkins.

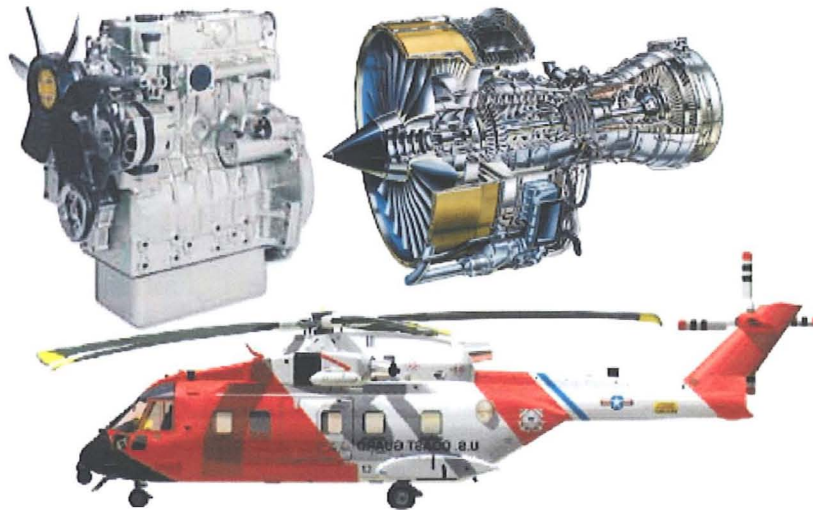


Fig. 4.14 EDC case studies: diesel engine, aero engine and helicopter

4.5.2 Studies outside the EDC

Extensive case studies have also been performed by other universities. While it was not possible to access the raw data for these studies, the resulting publications help confirm the generality of process planning challenges.

Ford

Yassine et al. (2001) explored the importance of iteration during the design of car doors. They found planning for design work that involved iteration was difficult and that more effort was justified up front to avoid iteration. They also found that some managers tended to hold on to resources as a contingency strategy for dealing with expected iteration work, an approach which sometimes led to inefficient use of resources. Although this work was performed outside the EDC, the opportunity arose to discuss the work with one of the authors (Dan Whitney). These discussions provided deeper insights into the study than could otherwise be obtained.

Intel

A study of new product development at Intel by Eppinger (2001) showed how the process can be improved by restructuring the tasks order. The study highlighted the importance of understanding connections between tasks and reducing iteration. An independent study at Intel by Nichols (1990) highlighted the importance of correctly managing engineering change: he noted that failure to manage change appropriately can result in delays to new product introduction. This observation has implications for the planning of projects that are highly susceptible to change, particularly with respect to the link between product and process.

Boeing

The management of different risks during the new product development process at Boeing was explored by Browning (1998). He used simulation to investigate how alternative task configurations could reduce the overall process risk. His model was later analysed by Cho and Eppinger (2005) to demonstrate the applicability of a more advanced simulation tool. Their model aimed to identify opportunities for process improvement by evaluating alternative process execution strategies.

Other studies

Several other studies have also been carried out in engineering companies. Clark and Fujimoto (1991) compared new product development programmes in Japanese and western auto manufacturers and observed the superiority of the former in planning and managing engineering changes in the design process. A study by Whitney (1993) at Nippondenso Co. Ltd. (a Japanese auto-component supplier) showed that increased flexibility can be obtained without sacrificing efficiency by improving the design process, in particular by simultaneously designing the product development and manufacturing processes.

As design-times decrease against the backdrop of increasing product-complexity and declining levels of overview and experience within organisations, the challenges of planning and managing large projects are likely to escalate. Even in the current design context, planning challenges are omnipresent in the companies studied.

Some interesting contrasts between Perkins and other companies were nonetheless noted – while both Perkins and Rolls Royce have trouble with planning iteration, Perkins seems to struggle mainly with rework, while Rolls Royce also has problems with iteration due to solution refinement and task repetition. At the same time, key Perkins employees have an excellent overview over the entire product while this is not true for Westland Helicopters or Rolls Royce. This discrepancy may be due to the different scale of the products being designed. Even within Perkins, however, project-scale emerged as a research-worthy topic.

4.6 Chapter reflection: opportunities to support design planning

Project planning in industry presents major challenges. Case-study data shows that the problem is multi-faceted, involving the interaction of product uncertainties and process unknowns. Project plans have many different applications in the hands of diverse stakeholders. They are used to scope projects, to plan budgets, to plan workloads, to track confidence in the design and to ensure that deadlines are met. As a result, plans are rarely optimised for all of their intended uses. Each application requires that different attributes of the design project be captured in the plans. Including everything in a single plan makes it cumbersome and difficult to use; omitting important information can lead to oversights.

In practice, many plans are used together to capture different aspects of the project. In addition to specific limitations of individual plans, problems were also noted with the concurrent use of multiple plans. Further, a gap was identified between the information captured by industrial project plans and the actual events that take place during new product development programmes. This gap is often bridged through the overview of experienced designers, an approach which is likely to cause problems as these employees retire.

Increasing product and process complexity highlight the limitations of current planning techniques. While Perkins acknowledges that it has problems with planning, it has difficulty discerning problem-causes from effects: it does not know what to change about the way that it plans because it cannot predict the impact of such changes. Sensitivity to task-failure and rework, as well as project scale and connectivity, is not well understood. Likewise, analysis of product-process interrelationships, from a confidence perspective, is proving to be theoretically useful but practically problematic.

Broadly speaking, the planning challenges in Perkins echo strongly those observed in the other case studies. The following issues were considered particularly relevant:

- 1) *modelling and representing uncertainty in design processes is a challenge: several different types of model may be used concurrently to capture different aspects of a process but co-ordinating these plans is difficult;*
- 2) *process sensitivity to scale and connectivity-level is frequently unknown or unclear;*

- 3) *the sensitivity of processes to iteration in general, and rework in particular, is a recognised issue but the impact of rework on process plans is poorly understood;*
- 4) *the implications for the process in response to changes to the product are difficult to predict – the use of confidence modelling to explore this issue is a research-worthy topic.*

These case study findings are supported by discussions with practitioners and academics at conferences and workshops. While some of the above topics (e.g. iteration and uncertainty) are considered in the literature, this study has probed deeply to explore how these issues are dealt with in practice and what challenges arise. It also drew attention to the importance of scale and complexity, and the product-process link, and illuminated the need for support tools to address the resulting planning challenges.

5

Foundations for Robust Plans

Previous chapters have highlighted concerns relating to process planning based on reviewed literature and on a detailed industrial case study. This chapter elaborates on key planning challenges from a theoretical perspective and lays the conceptual foundations for robust plans – plans that will result in processes that exhibit low sensitivity to different uncertainties.

To create robust plans, it is necessary to determine process sensitivity to such issues as rework, scale and task-connectivity level. Product-process interdependencies also warrant investigation. Effective modelling and representation of processes underpin all of these analyses. This chapter will define the requirements for modelling and sensitivity analysis of design processes with the goal of creating more robust plans. These requirements motivate the remaining thesis chapters.

5.1 Improved modelling of complex design processes

Chapters 3 and 4 highlighted the importance of modelling in understanding and improving the design process. This section discusses theoretical and practical factors which must be considered in constructing better design process models and improved plans.

5.1.1 Model building and evaluation

Design process model-building is a demanding activity due to different reasons. Firstly, such models depend on the perspective of different stakeholders involved in their construction, but there is no way of guaranteeing the impartiality of these individuals. In fact, it is likely that some of the information on which the model is based will be biased. Secondly, even if the information used to construct the model is completely unbiased, it is likely to contain errors because the model is constructed subject to assumptions and uncertain information. Oversights may also arise during the abstraction phase of model building, for example, due to incompleteness – even if a model is free from errors, it may still lead to poor results if important information is omitted. At the same time, good models apply the parsimony principle – they are as simple as possible while still achieving the desired functionality. Knowing what to include and what to omit can prove particularly difficult for complex systems where even small variations (e.g. those due to errors or oversights) in the input can have a major impact on the output.

Related challenges associated with model-building are verification and validation. Verification involves checking that the model meets its requirements and validation ensuring that it is fit for its intended purpose (Sargent, 1998). Many models contain probabilities – in the absence of rich data from previous projects, it is difficult to ensure that these values are correct. Data from previous projects is invaluable in validating the model's utility in solving a predefined problem(s). Nonetheless, the re-use of data from previous projects can be misleading due to changes in the design context. Another problem with model development through the modification of existing models is that legacy effects arise which can result in inefficiencies.

Notwithstanding the difficulties involved in process modelling, the activity itself can lead to valuable process insights. Signposting models, for example, typically contain information about the level of connectivity between tasks, the number of risky tasks,

duration uncertainty, resource constraints, schedule deadlines, rework characteristics and information about which tasks contribute the most to product confidence. This information contains many clues to the way in which a process is likely to behave – a highly connected process that contains a high percentage of risky tasks is likely to prove difficult to plan.

5.1.2 Representing uncertainty

Plans are subject to uncertainty and should hence cover a range of outcomes. Representing uncertainty in a process model presents challenges – especially when alternative routes through the design process (different task-orders) must be considered. Signposting models are particularly rich in terms of the number of process-routes which can be captured, a factor which makes their representation especially challenging. While a Gantt chart can represent a single route through the process, it does not accurately represent probabilistic information about task-ordering alternatives.

Challenges also arise when using DSMs to represent flexibility in task-order. Consider, for example, when different tasks, e.g. task C or D, can create the same information (in addition to other distinct information contributions), which is required for a dependent downstream task (e.g. task E). In this case, the flexibility of the task-order due to the logical OR connectivity (C-D-E, D-C-E, C-E-D, D-E-C) is not easily captured by a DSM.

Similarly, the order of tasks, which have not yet been executed, can affect the optimal choice of task-order at a given point in the process (see section 5.3.1 on rework). Because the exact task-order is often unknown prior to process execution, representing the connectivity between tasks is problematic. This poses difficulties when representing the process model even though it contains a lot of useful information about the way in which the process is likely to behave. In addition, the richness of data contained in probabilistic models means that a single representation often fails to adequately represent all relevant data.

5.1.3 Enhancing the Signposting model structure

The Signposting model provides a powerful way to explore design processes (Section 3.4). Nonetheless, enhancements to the model structure were required to model precedence-driven dependencies and parallel tasks that affect the same parameter.

Modelling task precedences

At Perkins, many task dependencies are based on hard precedences. For example, testing must follow procurement, although procurement does not explicitly contribute to design confidence. The absence of hard precedences in Signposting has previously been overcome by introducing artificial levels of confidence. However, this approach led to incompatibility between Perkins and Signposting models and increased the challenge of model-building because it required that Gantt-chart precedences be translated into confidence-based dependencies. This work identified the need for a more appropriate means to model task precedences. The development of a suitable solution is described in section 6.3.1.

Modelling parallel tasks

The previous Signposting model (O'Donovan, 2004) defined discrete output states for each parameter, based on a given input state. This led to problems when modelling parallel tasks which simultaneously influence the same parameter, as illustrated by the following example. Consider a design which undergoes two independent tests, Task A and Task B, both of which affect parameter 1 and at least one other parameter (Fig. 5.1). It is the parameter(s) other than parameter 1 which distinguish the tasks.

Let us assume that the first task, Task A α can be executed when the confidence in parameter 1 is at least 50% and leads to an output confidence of 55% in parameter 1. Let us further assume that an alternative version of the first task exists: Task A β can be executed when the confidence in parameter 1 is at least 55% and leads to an output confidence of 60% in parameter 1 (Fig. 5.1a).

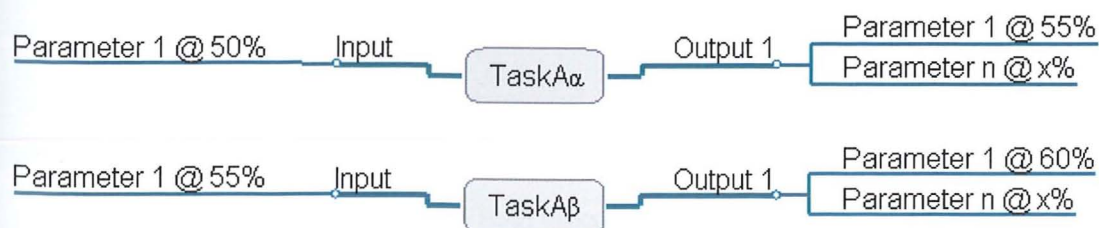
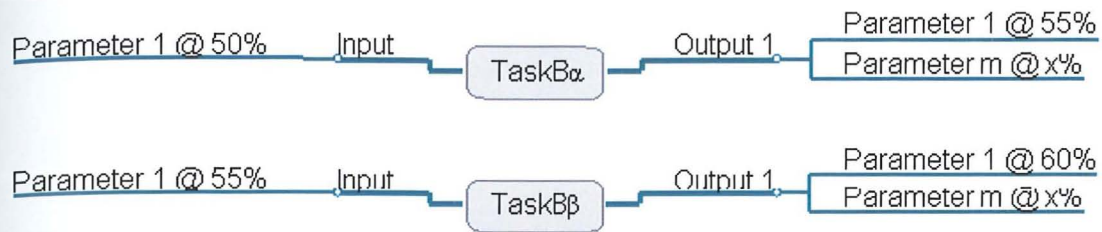


Fig. 5.1a Representation of Tasks A α and A β

Fig. 5.1b Representation of Tasks $B\alpha$ and $B\beta$

Let us make the same assumptions for Task B: Task $B\alpha$ can be executed when the confidence in parameter 1 is at least 50% and leads to an output confidence of 55% in parameter 1. Let us again assume that an alternative version of the task exists: Task $B\beta$ can be executed when the confidence in parameter 1 is at least 55% and leads to an output confidence of 60% in parameter 1 (Fig. 5.1b).

As the process, which contains tasks A and B, is executed a point is reached when the confidence associated with parameter 1 reaches 50%. If both tasks are performed in sequence – for example, due to resource constraints, the output confidence is 60% (Fig. 5.2). If, however, both tasks are performed in parallel, the resulting confidence for parameter 1 is 55% (Fig. 5.3), an incorrect value which arises because the model fails to account for the fact that both tasks contribute confidence to the same parameter simultaneously. Because each of the parallel tasks only considers the parameter confidence at input when calculating the output confidence and hence fails to take into account how a parallel task can affect the same parameter, the model effectively ignores the confidence contribution to parameter 1 from the shorter of two parallel tasks. This confidence contribution is overwritten upon completion of the longer parallel task which affects the same parameter. If both tasks finish simultaneously, the starting order of the tasks will determine which confidence mapping is overwritten. This limitation of the model constitutes a research opportunity for this work.

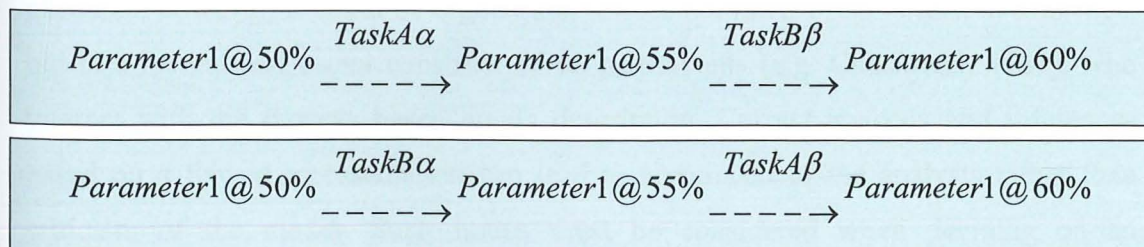


Fig. 5.2 Both tasks in sequence

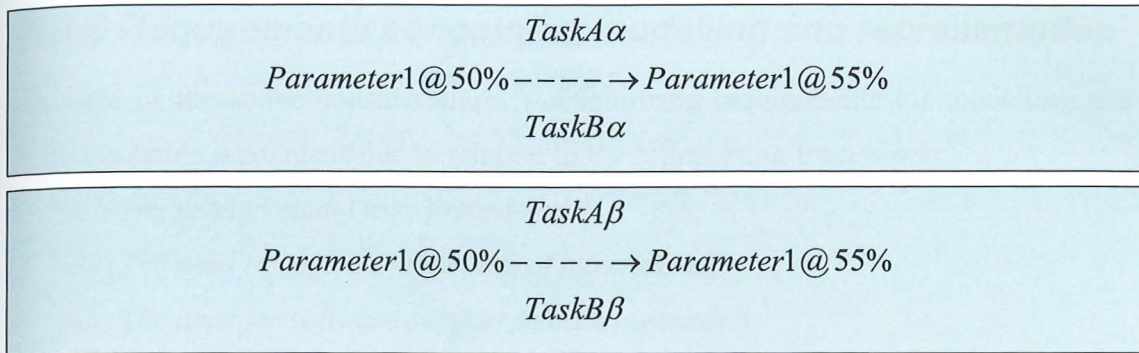


Fig. 5.3 Both tasks is parallel

5.1.4 Separating model analysis from process analysis

Analysis of design processes is usually performed using process models. Different models (Chapter 3 and 4) contain different information and have different affordances: the content of the model affects the way in which it can be used to reason about processes. For example, conventional DSMs show how tasks are connected together but do not give any information about task-durations. Gantt charts represent both task-durations and task-precedences but do not show iteration. Thus, Gantt charts are useful when trying to predict process-durations while DSMs provide useful insights into process-connectivity. Alternatively, Signposting contains information about both the product (in the form of design parameters) and the process and hence outperforms other models when making product/process trade-offs (Section 5.4). At Perkins, several different models were used to plan and manage the design process (Chapter 4). Each model has different merits and limitations and the choice of model affects the way in which issues can be explored and constrains the insights that can be obtained.

Choosing an appropriate model is further complicated by the fact that plans are created at a point when the process does not exist. Even when a process has been executed, however, it could be argued that it does not have a real objective existence, but is a mental and social construct of its participants (e.g. Checkland, 1981), who interact with the process based on its description. Correct analysis and inferences based on a flawed representation can lead to scepticism of the analysis rather than criticism of the model. Such issues must be considered when deciding on an appropriate model for the design process.

5.1.5 Requirements concerning modelling and representation

In light of the above considerations, the following requirements for modelling and representation were identified in relation to the Signposting framework:

- M1) The need to model task-precedences;*
- M2) The need to improve modelling of parallel tasks;*
- M3) The need for software support in creating models;*
- M4) The need for conventional task-based process representations;*
- M5) The need for process visualisations (e.g. DSMs) to represent the connectivity.*

5.2 Investigating the effects of process structure

Processes vary both in terms of their structure (scale, connectivity-level, rework-behaviour) and in terms of the uncertainty associated with specific details such as task-durations, task-outcomes and resource availability. For example, the design of an aero-engine is different from that of a diesel engine due to variations in scale and iteration, even though both design processes are susceptible to test-failures and task-delays. Analyses are required to separate process-structure effects from complexity due to uncertainty.

This situation is analogous to traffic congestion, where delays can be predicted by looking at structure of motorway interconnections (delays are likely where motorways merge) or alternatively by examining specifics such as number of vehicles, accidents and road-works (Johnson, 1983). Ideally, one should consider both structure and specific information when planning a journey.

Simulation analyses can be used to explore how different uncertainties affect plannability and how variations in structure influence project-performance during execution. Chapters 7 and 8 of this thesis will show how insights into both structural variations and specific sensitivities are useful in creating more robust plans. This section discusses the theoretical issues which underpin these analyses.

5.2.1 Exploring the effects of process-specific variations

Process levers (see Section 1.1.2) can be identified by modelling different task-outcomes and investigating the corresponding variation in process duration. While it is obvious that delays to any activities along the critical path will lead to process overrun, the challenge lies in identifying nominally non-critical tasks which become

critical in the event of task-failures/delays. In Figure 5.4, the critical path is highlighted in red (Tasks 2,3,4 and 5). However, there are two other almost critical chains (Tasks 2,7,10 and Tasks 2,8,9) which may become critical in the face of uncertain task durations. Task 2 is the only task which is common to all three possible critical paths; it will act as a process lever regardless of which path eventually proves critical. In contrast, many of the other tasks are close to the critical path and will act as process pitfalls should they be delayed.

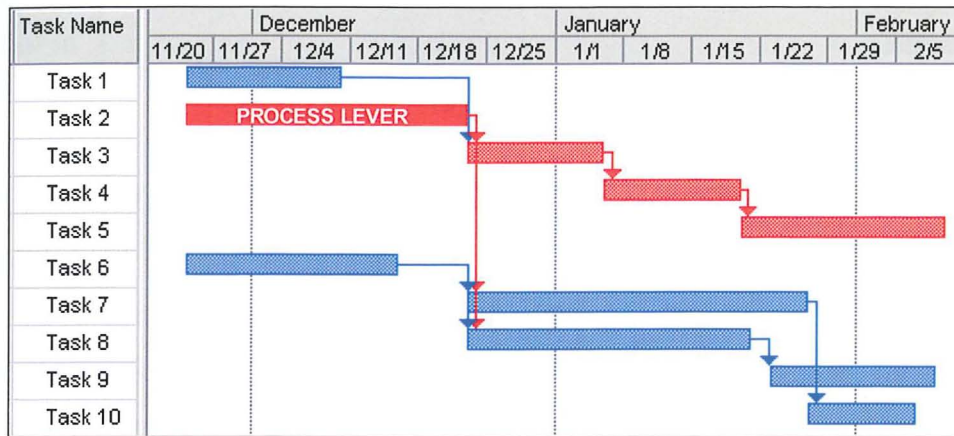


Fig. 5.4 The second task acts a process lever

Problems such as unplanned iteration can also cause non-critical activities to become critical. For example, if technical drawings for a support bracket are not completed on time, CAE testing may be performed based on preliminary information. The resulting problems may only be detected much later during physical component testing and the resultant rework leads to delays. In this case, the technical drawing activity can be seen as a leverage mechanism for process success. Similarly, combinations of failures/delays to non-critical tasks can also lead to delays.

As tighter deadlines reduce the time available for new product development programmes, companies are faced with the challenge of identifying the most appropriate strategy for process-acceleration. Process analysis can help them identify bottlenecks and thus prioritise opportunities for process improvement: acceleration of tasks, which take place at process bottlenecks, has a greater impact on the process than reducing the durations of critical tasks which are performed in parallel to other almost parallel tasks (Fig. 5.5). Such acceleration can sometimes be achieved through the re-prioritisation of resources.

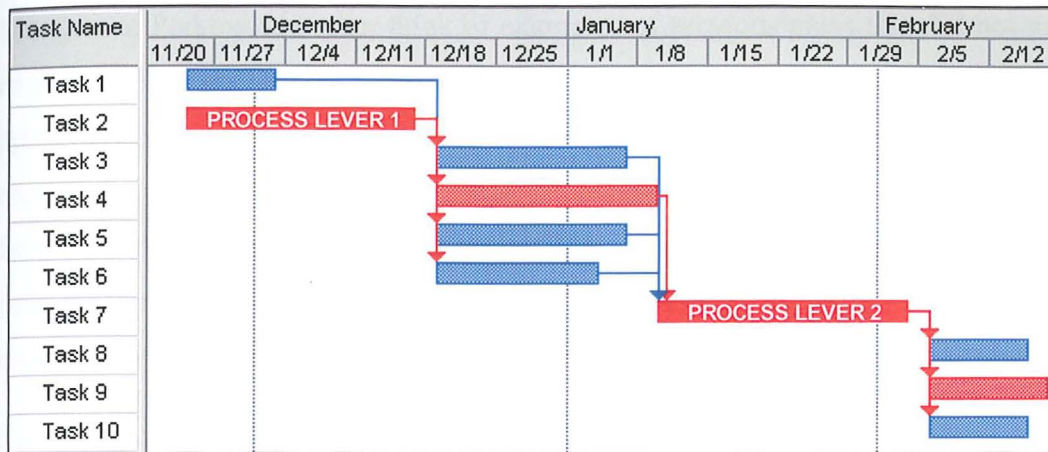


Fig. 5.5 Process levers at process bottlenecks

5.2.2 Exploring the effects of structural variation

In addition to specific uncertainties, the implications of process variations at a structural level also warrant attention. Much process analysis in academia has focused on small, simple models while, in reality, many models are complex. Intuitively, it makes sense that planning large-scale design processes is more difficult but the issue has been insufficiently addressed in the literature. Likewise, process-sensitivity to task-connectivity warrants further attention, both in terms of the number of connected tasks and the patterns of connectivity.

Other variations in process structure are also important. Even when two processes have the same scale and connectivity level, variations due to the timing of high-risk tasks may impact process performance. When rework takes place early in the design process and is discovered quickly, major process delays are less likely. In contrast, task failures late in the process are likely to prove problematic.

Similarly, the degree of connectivity may influence the number of tasks which require rework in the event of a given task-failure: highly connected processes are likely to suffer more severely. Also, the time taken to discover rework may be influenced by both the scale and connectivity-level of the process: in simple processes, a single designer may be responsible for all of the design tasks and thus quickly identify mistakes; in a complex process, a wide range of issues (scale, complexity, cultural differences, lack of overview) may cause mistakes to go unnoticed. Finally, the structure of the process, particularly in terms of connectivity, may provide clues towards the robustness of the process in terms of its ability to absorb rework.

Managers at Perkins can often think of examples of previous plans that did not work properly and expect that similar problems will arise with current plans. By clarifying appropriate expectations for different plans at a structural level, some of this scepticism could be reduced and different plans could be handled appropriately. Planners and managers would be better placed to identify similar processes and to draw appropriate inferences from other processes. They would also be placed in a better position to define appropriate changes to the process structure.

5.2.3 Requirements concerning model variations

The following requirements were identified concerning structural and process-specific variations:

- S1) The need for a means to explore structural variations of models without being constrained by process specifics;*
- S2) The need to explore structural variations between models in terms of*
 - a. scale;*
 - b. connectivity-level;*
- S3) The need to explore process-specific variations in an industrial model.*

5.3 Rework impact mitigation

Ensuring that processes are robust in the face of iteration in general, and rework in particular, is challenging. Difficulties arise in predicting which tasks will require rework, when the rework will be discovered, the duration of the reworked tasks and the task-order during rework. This problem is exacerbated by the fact that some rework is disguised as task delays and some managers deliberately fail to disclose known rework for personal or political reasons (Ford and Sterman, 2003). At the same time, planning for, and successfully carrying out, rework can be critical in process success. This section considers different approaches to mitigating the undesired effects of rework and iteration.

5.3.1 Accounting for likely rework

The first clue to the impact that rework is likely to have on a process is the rework-probability associated with different tasks. Both the number of high-risk tasks and the failure-probability associated with these tasks affect process behaviour. Analyses are needed to explore process sensitivity to task-failure probability.

Other factors too can influence rework. In some cases, process duration can be reduced by using an appropriate task order which considers the consequences of different task failures (Melo, 2002; Flanagan et al., 2005b). Some task-failures drive rework in other tasks, while others only require that the failed task be reworked.

A simple example helps explain how prediction of the type of task failure influences the level of rework required. Consider the design of two components A and B. Task B can create information which drives rework of task A. If task B fails to achieve a satisfactory outcome, rework may be required in both tasks; this is not the case for task A. Thus, performing task B before task A is likely to reduce the total rework associated with the process (Fig. 5.6). In both situations, task B fails. However, due to the improved task order, the resulting rework is reduced in the more robust plan as task A is performed only once. Hence, it is not sufficient to consider the likelihood of task-failure as an indicator of process-risk; the resultant rework must also be taken into account.

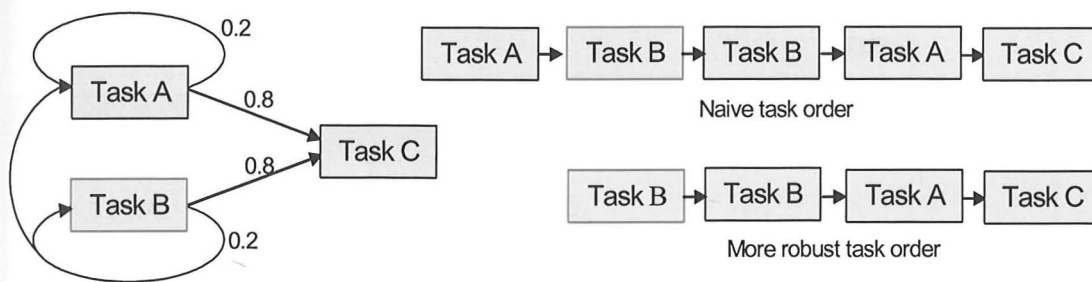


Fig. 5.6 Task order should reflect task failure mode

The above example shows how task-order can affect the resultant rework. Other factors too, may play an important role – for example, the duration of reworked tasks. Some reworked tasks become critical and are expedited by resource reprioritisation. Others are reworked quickly because the designers have learned from previous iterations. It would be useful to explore the effect of task-acceleration during rework.

5.3.2 Understanding process-sensitivity to task-failures

If rework is discovered late in the process, numerous tasks may have been performed subject to incorrect input information and much rework may be required (Cooper, 1993). In contrast, early discovery of rework avoids this problem (Fig. 5.7). Analyses are needed to investigate how the downstream task-order changes in response to undiscovered rework.

Not all task failures have the same implications for the process. Some failures act as 'show stoppers' – indeed, failure of an engine block late in the process could have massive financial implications for a diesel engine design program. Other failures have a much lesser impact – for example, failures of tasks, which are not on the critical path and are quickly discovered and reworked, often have a negligible impact on the process. Yassine et al. (2001) contend that it is sometimes worthwhile taking more time early in the design process to avoid rework later on but guidelines regarding which tasks to focus on are elusive. A means to identify tasks, which have a major impact on the process through rework, would be useful.



Fig. 5.7 Early and late rework discovery and resulting process delays

5.3.3 Requirements concerning rework

The following requirements for rework mitigation were identified:

R1) The need to explore process-sensitivity to rework probability variations:

- a. at a structural level;
- b. at a process-specific level.

R2) The need to explore the effect of task-acceleration during rework:

- a. at a structural level;
- b. at a process-specific level.

5.4 Exploration of the product-process link

Process risk is growing due to increased concurrency coupled with reduced overview and experience (Flanagan et al., 2006). There is a need to ensure that product quality does not suffer as a result. Process schedule risk is inversely related to confidence in the product quality but the link between these issues is poorly understood.

As schedule and financial objectives become tighter, a means to explore how changes to plans are likely to impact product quality is becoming increasingly critical. Currently such considerations rely heavily on the tacit knowledge of highly experienced engineers; as novice designers and contractors replace more experienced personnel, this issue is particularly relevant. Even for experienced engineers, support in assessing product-process interdependencies would be appreciated.

As noted in the previous chapter, Perkins has created a mapping which shows how different tasks contribute to confidence in different components. However, this data is not represented graphically by the company. This section provides the theoretical foundations for confidence-based process-analysis.

5.4.1 Signposting confidence versus Perkins confidence

The notion of confidence in Signposting has existed for almost a decade but succinct definitions of the terms are sparse. Nonetheless, at least two definitions do exist. Stacey and Eckert (2003) define confidence *"as the degree to which an aspect of the design can be relied on as satisfactory"*, adding that *"confidence defines the expected stability of the space of possible designs."* Melo (2002) defines confidence as *"the refinement that a designer considers a parameter to have, at a point in the design process."* When applying these definitions, a high parameter confidence indicates that the parameter will not be changed during the rest of the design, that the design satisfies the requirements, that the design is of a sufficient quality and that the design work is almost complete.

Clarkson and Hamilton (2000) and Melo (2002) limit confidence to four levels: high, medium, low and zero. O'Donovan (2004) found this view of confidence overly limiting and replaced the notion of confidence with that of parameter qualifiers which are usually intended to reflect the maturity of the parameters at different points in the design process.

The advantage of parameter qualifiers is that they allowed the definition of countless different states for a single parameter. If 10 different tasks affect a parameter then the parameter state varies from 1 to 10 as the design progresses. Alternatively, if only 5 tasks affect another parameter, this parameter reaches a state of 5 upon design completion. Thus, confidence was changed from an absolute to a relative measure. The same number could convey different meanings (5 means completion for the second parameter but not for the first) and the original utility of confidence, as a mechanism to convey subjective designer belief about the maturity of different parameters was largely lost. Instead, it is used as a mechanism to differentiate different stages in the process.

In all versions of Signposting, confidence (or parameter qualifiers) is associated with parameters. Examples of parameters used by Clarkson and Hamilton include stress, loads and geometry. However, O'Donovan's interpretation of parameters is more extensive and includes such examples as ergonomics and aesthetics. The definition of such parameters is more subjective than that of parameters such as stress and loads which are associated with objective, numerical values (O'Donovan, 2004).

Confidence, as used in Perkins, is similar to the original Signposting confidence, in so far as it conveys an impression of the designer's belief in design maturity, but also incorporates O'Donovan's notions of multiple levels and progress through a process. However, while Signposting confidence is typically used to model abstract performance-type parameters, confidence at Perkins is used only to describe confidence associated with components. Also, while tasks in Signposting models require a predefined input level of confidence, precedences, as opposed to confidence levels, are used to guide task-choice in Perkins – only task output contributions to confidence are defined. Finally, the use of confidence in Perkins is pragmatic – while the company acknowledges the existence of theoretical difficulties in defining and using confidence (Section 4.4.4), it is prepared to ignore these in light of the practical benefits which can be obtained.

5.4.2 Confidence-based process-analysis

Used in conjunction with Gantt charts, confidence-data could give designers a better overall picture of the design process in terms of the tasks that influence confidence growth. Confidence-based analysis could also show how combinations of task-failures and delays are likely to affect parameter confidence values. Such knowledge would be

beneficial in prioritising different tasks. Confidence data also provide a means to compare and contrast alternative process configurations because the same parameters are present in all process alternatives, although different tasks, such as CAE analysis instead of physical testing, may be used to grow confidence.

The timing of confidence growth for different components varies; some components mature early in the design process while others accumulate confidence at the project conclusion. Examination of confidence data shows which parameters mature late in the process – if the associated tasks fail or overrun, delays are likely. Conversely, parameters which mature early on are less likely to cause problems. In some cases, the data suggest opportunities to reduce risk by gaining more confidence earlier in the process (by reconfiguring the order of activities).

Processes which appear similar from a task-based perspective may differ considerably in terms of the confidence growth associated with different parameters. At the same time, comparatively large changes to the task configuration may have little impact on confidence growth profiles. Configuration changes arise, for example, when a task can have multiple outcomes. In such cases, the task-based view only shows which tasks have been completed but does not reflect resulting changes to design confidence. In contrast, confidence data highlight variations due to different task-outcomes.

Confidence data provide valuable information about the parameter's development through the process and are useful when trying to appreciate the significance of a specific task in the context of the overall design.

Together with Gantt charts, confidence data can provide insights into process behaviour and answer several key questions such as:

- *what is the projected confidence level associated with the parameter at different points in the process, based on the plan?*
- *which tasks affect this parameter?*
- *how will delays to different tasks affect this parameter?*
- *when does the parameter confidence grow?*

Despite the potential utility of confidence, its application to process-analysis within Perkins is limited. One possible reason for this is that confidence data is not linked to data on Gantt-chart plans. Also, tools for confidence-visualisation and confidence-based process-analysis are sparse.

5.4.3 Confidence and rework

As process scale and level of task-connectivity rise, there is an increased chance that a parameter's confidence will fall due to combinations of task-failures. Spotting these combinations of failures can be very important, especially in scenarios where the process is reasonably robust against a single failure but cannot recover easily from multiple failures which affect the same parameter. For example, if the CAE analysis cannot be completed such that it fails to identify a design weakness, and this weakness is also missed during rig testing, then the problem will only be discovered during engine testing. At this late stage in the design process, corrective action will be expensive and/or will result in delays. However, if either CAE analysis or rig testing identify the problem, a potentially expensive delay can be avoided. The use of confidence data together with Gantt charts can highlight which combinations of task-failures are likely to be problematic and can help designers and managers develop appropriate actions for process-risk mitigation.

Similarly, during process execution, confidence targets based on scheduled work could be compared against confidence levels resulting from actual progress. Progress in the task domain could be assessed by checking which tasks were completed. However, it would also be possible to determine progress from a product confidence perspective, and thus to identify which tasks have delivered their expected confidence contribution. Gaps between predicted and actual confidence, despite on-time task completion, point to downstream rework. By helping to identify tasks which have failed to deliver their expected confidence contribution, confidence data place management in a better position to make predictions about delays to downstream tasks and to implement rework mitigation strategies. Again however, there is a need for improved tools to assist management in exploring the sensitivity of design confidence to rework.

5.4.4 Requirements concerning the product-process link

The requirements relating to the product-process link are as follows:

- C1) The need for a Signposting tool that can deal with Perkins' interpretation of confidence;*
- C2) The need for a new representation that shows how confidence grows during process-execution;*

C3)The need to concurrently examine the process from a task-based and from a confidence perspective:

- a. at a structural level;*
- b. at a process-specific level.*

C4)The need to explore the link between confidence growth and rework:

- a. at a structural level;*
- b. at a process-specific level.*

5.5 Summary

This chapter defined requirements for creating robust plans in terms of process modelling and representation; scale and connectivity; rework, and product-process dependencies. The next chapter outlines a solution concept, which meets these requirements, and its embodiment.

6

Developing a tool for design process analysis

So far, this thesis has elucidated the challenges associated with design planning and outlined the requirements for a means to address these challenges. The proposed solution should provide assistance for modelling and allow more comprehensive exploration of the way in which factors such as rework, scale and connectivity affect the robustness of plans. In line with the challenges highlighted in the industrial case study, it should also facilitate investigation of the product-process link. This chapter proposes the development of a software tool that meets these goals and facilitates the analysis of real-world and hypothetical models. It proceeds to detail its embodiment and concludes with a section on tool-verification.

6.1 A solution proposal for design process analysis

In light of the requirements outlined in the previous chapter, a concept was conceived for the exploration of design processes which involves the analysis of both real-world and hypothetical models (Fig. 6.1). Both sets of analyses can be used to explore how processes are likely to behave: hypothetical models can yield valuable, theoretical insights into the way in which process-structure impacts performance, while real-world models are better suited to analysing the effects of project-specific variations. A more thorough discussion of hypothetical models is provided in section 6.2.

Regardless of the type of model being considered, this work proposes a six-step analysis approach (Fig. 6.1). Firstly, the model is visualised so that connectivity between different model variables can be examined. Next, the model is perturbed, such that sensitivity to different factors can be explored. By considering different perturbation visualisation, further insights into the likely process behaviour can be obtained.

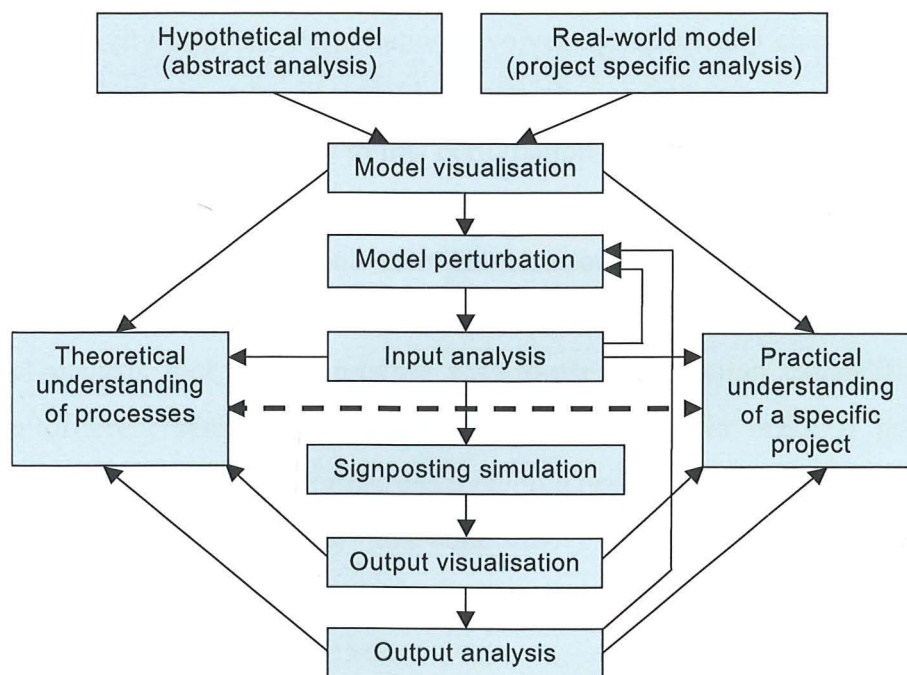


Fig. 6.1 The solution concept

Following analysis of the input model, Signposting simulation analysis is performed. The simulation results constitute a considerable volume of data – improved visualisations are required to facilitate analysis of this output data and hence to better understand process behaviour. Simulation output visualisation and analysis are the final steps in the proposed solution approach. However, the analysis approach is

iterative and insights from simulation output may be used to guide further perturbations and future analyses.

The remaining sections of this chapter reflect the structure of the above diagram (Fig. 6.1) and concern the development of a software tool which embodies this solution concept. Firstly however, the specific requirements for this tool are defined.

6.1.1 Specific requirements for concept embodiment

Requirements defined in the previous chapter concern the need to explore the effects of scale and connectivity (Requirement S2), rework (Requirements R1 and R2) and the product-process link (Requirement C3). However, the specific requirements for a tool to perform such analyses have not yet been defined. These requirements concern model-perturbation, Signposting simulation and visualisation of the simulation output as discussed below.

Develop perturbation capabilities

Model perturbation, together with simulation, provides an effective means to perform process sensitivity analysis. Perturbation involves systematically changing the value of a given variable or group of variables. Once the model has been perturbed, it can then be simulated and sensitivity to this perturbation can be determined by comparing results from the original and perturbed models. In this manner, project sensitivity to several different uncertainties and their combinations can be established, providing useful insights into how project properties affect plannability.

A model-analysis tool should provide easy-to-use perturbation capabilities which facilitate different sensitivity analyses. Such a tool would make it possible to systematically perturb process properties including task-duration, task-duration during rework, task-failure likelihood, task-connectivity level, model-scale, and task-contribution to parameter-confidence.

Develop simulation capabilities

Simulation (Section 3.3.1) lies at the heart of the sensitivity analysis performed for this thesis. Hence, the simulation software should be reasonably fast and easy to use; otherwise the approach is likely to be resisted by practitioners. The simulation code should also account for any changes to the underlying modelling framework, e.g. the modelling of task-precedences (Section 5.1.3).

Previous versions of Signposting use a simple algorithm to pick tasks based on the information level associated with the design, at the time when the task is being chosen. However, it does not consider the likely information contributions of the tasks available for selection when choosing the most appropriate task from a list of available tasks. Improved algorithms for picking tasks are required for a Signposting simulation tool that better reflects industrial practice.

Different algorithms should be developed to improve the way that tasks are chosen during the Signposting simulation. A task could be chosen based on an average output state (computed as a weighted sum of different possible outcomes), based solely on the most likely outcome, or based on the best possible outcome (a highly optimistic approach). In any case, the likely task outcome should be given greater consideration during the simulated task-execution.

Other improvements to the simulation code are also required, not least of which is increased simulation speed. Simulation analysis using the previous Signposting code was highly time-consuming taking on the order of days for medium-sized models. Ideally, real-time simulation analyses are needed; achieving such improvement would greatly boost the appeal of the tool and would encourage users to explore multiple different model perturbations using simulation.

Create new representations for simulation outputs

Simulation analysis produces extensive data on potential process behaviour – while this data can provide useful insights into how the process is likely to unfold during execution, its interpretation in text format is tedious and error-prone. New visualisations are needed to provide graphical and statistical summaries of the simulation results. In addition, multiple views can be beneficial in facilitating analysis of rich data (Packham and Denham, 2003) – multiple views of Signposting simulation results should be provided. Further, different views of the same process should be interlinked to facilitate process analysis. Because Signposting simulation data results in several alternative routes through the process, an effective means to select a representative process from a number of simulated runs would be beneficial. Visualisation of the simulation output was extremely limited in earlier versions of Signposting; opportunities to address this limitation were manifold.

Future users of the tool (both industrial practitioners and academics) may require different representations specific to their product or development programme which

cannot be envisaged during the software development phase. While several conventional visualisations (Gantt charts, DSMs, matrices, tables) should be included in the simulation software, a means should also be provided to interact with standard data analysis packages such as MS Excel™.

Commercially available software packages can perform simulation analysis for tasks subject to uncertain durations and costs. They do not, however, provide any information on how these uncertainties are likely to influence product confidence. A visualisation of variation in product confidence in response to changes in task-timing could prove especially relevant for inexperienced designers who have limited understanding of how failure to complete their tasks to the desired level of confidence will impact others downstream.

The following specific requirements were identified for a modelling and analysis tool which can be used to explore rework, scale and connectivity and the product-process link.

T1) the need for model-perturbation capabilities which facilitate sensitivity analysis with respect to:

- a) task-duration variation during rework;*
- b) task-failure likelihood;*
- c) task-connectivity level;*
- d) model-scale;*
- e) task-contribution to parameter-confidence;*

T2) the need to improve the execution speed of the simulation code;

T3) the need for better algorithms for picking tasks;

T4) the need for new visualisations of model data and simulation results which focus on:

- a) a task perspective (e.g. Gantt charts, DSMs);*
- b) a confidence perspective;*
- c) a combined product-process perspective;*
- d) multiple interlinked views of the process in combination;*
- e) representative process runs.*

6.2 Analysis of hypothetical design process models

This section introduces hypothetical models – a core thesis topic which draws from and builds on research into modelling and representation, and underpins the analyses into process-structure, rework and the product-process link, described in Chapter 7. The use of hypothetical models allows the separation of the structural problems from specific ‘what-if’ questions (Fig. 6.2). In addition, the analysis of hypothetical models can lead to improved understanding of the manner in which different process properties affect plannability. Hypothetical models can be used to explore a wide range of processes with different characteristics, while case study data usually focuses on a very small number of processes. The use of hypothetical models also avoids the problems of establishing how well a model characterises a real project - it avoids the issue of separating modelling errors from simulation errors while acknowledging that both can lead to erroneous results.

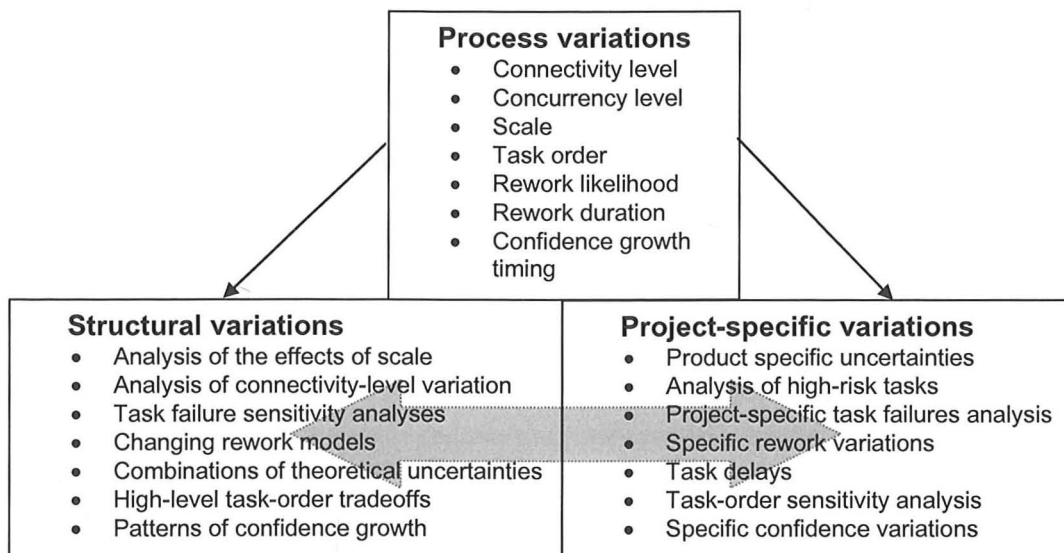


Fig. 6.2 Using hypothetical and project-specific models in combination

6.2.1 Advantages of hypothetical models

Hypothetical models allow consideration of project properties and uncertainties independently of the constraints of any real-world projects. This overcomes concerns about bias during the interpretation of simulation analyses. In a sense, the use of hypothetical models allows the user to think outside the box – in contrast to real-world process analyses which start with a host of preconceptions and assumptions concerning the nature of particular model.

The way a project is described and modelled profoundly affects how the risk is assessed and the project carried out (Section 5.1.1). Using hypothetical models, it is possible to look at the structural properties (Section 5.2.2) of models – in terms of degree of detail, number of iterative tasks, and number of parallel tasks to explore their effect on project risk and plannability – without being drawn into discussion about how well the model represents an actual process (Requirement S2, Section 5.1.5).

The availability of a large number of distinct models is advantageous during simulation tool development. Different models are needed to test software and ensure that software bugs are discovered and removed prior to industrial application. Generated models can be used to test new software functionality. Software features can be developed and tested using generated models and then validated based on case study data concerning actual projects.

As stated earlier (Section 5.1.3), the challenge of model-building acts as a barrier to simulation in industry. The use of generated models avoids this problem; as the effort in generating the models is considerably reduced, it is not necessary to demonstrate the same level of benefit from the results in order to make the approach practical. Even if only a few practical implications can be inferred based on the generated models, the ratio of benefit over time invested, is high.

Despite their potential utility in exploring structural variations between processes, hypothetical models are not well suited to examining process-specific variations. To this end, real-world models are needed. The next section discusses the elicitation of industrial process models.

6.3 Building real-world process models

Signposting models are rich in data: they require information about task properties as well as parameter confidences. Previously, the effort involved in constructing Signposting project models has proved a barrier to the industrial acceptance of the technique. As part of this work, several new features were implemented in the modelling tool to address this limitation. In addition, changes to the Signposting model-structure were realised which aim at increasing its applicability for analysing industrial design processes while nonetheless simplifying the model-building process. These changes, which concern modelling hard-wired task-precedences and parallel

tasks, are also reflected in the Signposting tool. The realisation of these changes results in a Signposting tool which can deal with Perkins' interpretation of confidence, thus satisfying requirement C1 (Section 5.4.4).

6.3.1 Modelling task precedences in Signposting

To address requirement M1 (Section 5.1.5) – the need to model task precedences in Signposting – this work introduced the concept of task-precedences into Signposting. This improved the flexibility of the tool and simplified the modelling process, while retaining the confidence concept. Task precedences are captured through the use of a new type of parameter, the precedence parameter. Such parameters store binary information about whether or not a task has been performed. When the task is executed, the state of the precedence parameter changes from zero to one. As the design process progresses, each downstream task checks the status of precedence parameters in its input state; it cannot be performed unless the required predecessors have been completed.

Although other approaches for modelling precedences were considered, this approach was chosen because it has a minimal impact in terms of changes required for the simulation engine. At the same time, it constitutes a considerable conceptual change to Signposting's process modelling capability: the task order can now be chosen based on a combination of task precedences and parameter confidences. The degree to which either approach (precedences or confidences) is used is at the discretion of the model-builder. If desired, a model based entirely on either of these approaches is also possible, thus increasing the tool's flexibility.

Further, both precedence and confidence can be used together to guide task-choice during process execution: the precedences constrain the available choice of possible tasks and the confidence parameters provide useful guidance on the most suitable task to select for execution (Fig. 6.3).

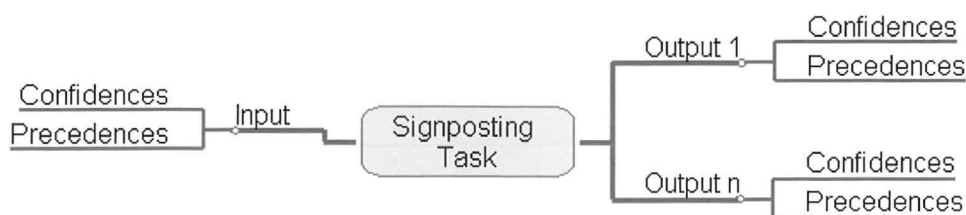


Fig. 6.3 Signposting task with confidence and precedence working together

6.3.2 Modelling parallel tasks using Signposting

In line with the requirement M2 (Section 5.1.5) – the need to improve modelling of parallel tasks in Signposting – further changes to the Signposting model structure were implemented.

In contrast to the old approach which defines the output directly, the new mappings specify the amount of confidence that should be added based on the given input state (Fig. 6.4). Thus each task is assigned a confidence contribution: e.g. the rig test provides a 5% increase in confidence as opposed to causing confidence to reach 60%.

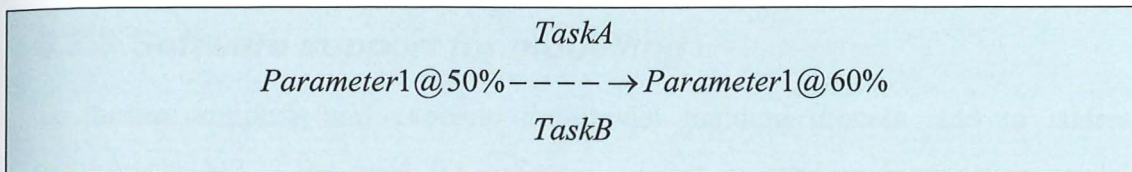


Fig. 6.4 Confidence contributions rather than discrete output states can model parallel tasks (If tasks A and B are performed in parallel, the confidence increases from 50% to 60%)

The confidence contributions may also be specified as context dependent: e.g. if the current parameter confidence > 10% and < 20%, add 10%, else add 15%. Several different confidence conditions can be applied to reflect different task execution contexts. In addition to overcoming the problems with simultaneous task executions, this approach simplifies the modelling process, as a vast range of possibilities are covered by a single confidence condition. For example, the condition: if parameter confidence > 40% add 5%, covers 45=>50%, 50=>55%, 55=>60% and a host of other possibilities (Fig. 6.5) while separate mapping for each condition would be required in the old modelling framework. The use of confidence contributions rather than discrete confidence outputs also aligns more closely to the way in which Perkins deals with design confidence.

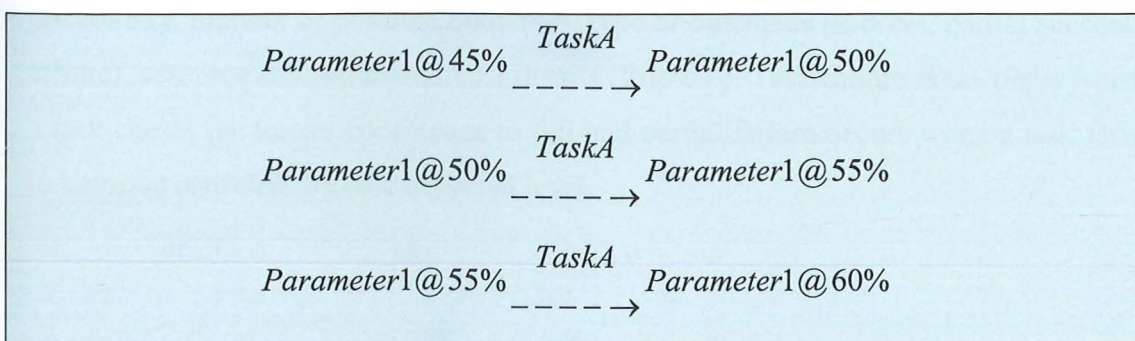


Fig. 6.5 One confidence condition covers several output states

Although efforts were made to keep the modelling framework as close to previous versions of Signposting as possible, the addition of confidence conditions was considered worthwhile as it improved the functionality of the model while reducing the model-building effort. Further, the capability of Signposting to model tasks using the original approach of discrete parameter confidence output values has not been overwritten: if the person building the model desires discrete values rather than confidence contributions, the extended modelling framework continues to deliver this functionality.

6.3.3 Software support for modelling

To further simplify and expedite the model building process, and to address requirement M3 – the need for software support in creating Signposting models (Section 5.1.5), new functionality was added to the Signposting tool, which allowed it to interface with MS Project files. Task data, such as resources and duration, could then be automatically extracted from the MS Project files and combined with confidence data to form the basis of the Signposting model. The introduction of hard-wired task precedences in the Signposting model increased its compatibility with existing industrial models, such as Gantt charts, and thus helped reduce the effort involved in model-building.

Although further data concerning task-risk (duration uncertainty, failure probability, resource variability) cannot be automatically added at the current time (as it does not exist within other company databases and requires expert knowledge (Chapter 4)), new interfaces have also been added to the software tool to simplify this phase of model building. These interfaces consist of dialogue boxes which allow the tool-user to enter/edit information concerning different task characteristics such as duration uncertainty, number of possible outcomes, type of outcomes (success, partial success, failure), resource utilisation and cost details (Fig. 6.6). Task-failure takes place when a task causes parameter confidence to fall and partial failure occurs when a task fails to increase confidence to the expected level.

Process Input Window

Model Type: Confidence Model

Max Cost: 100000.0 Max Duration: 100000.0

Target Cost: 100000.0 Target Duration: 100000.0

Buttons: GoalStates, Parameters, Resources

Task Name
7 Piston_Bowl_design
8 Current_casting_Cam_design_15%_(incl_procurement)
9 Lash-up_Twin_Turbo_arrangement_(incl_procurement)
10 Specify_BOM
11 Set_up_Mechanical_BOM

Task Input Window

Task Name: Piston_Bowl_design

Input Output Resource

Buttons: Add, Delete

OutputState1

Cost Mean: 1.0 Duration Mean: 35.3541666666715

Cost Variance: 0.0 Duration Variance: 0.0

Cost Uncertainty: 0.0 Duration Uncert...: 0.0

Probability: 1.0 Confidence Co...

Parameter	Value
Result of Piston Bowl design	1

Buttons: OK, Cancel

A	B	From Engine...	Fr
Engineering_activities_start-up_meeting	Input	☐	
973-High_power_rating	Input	☒	
Thermal_Analysis-High_Power_Rating	Output	☐	
Contingency-Other_ratings_(HP)	Input	☐	
Predictive_modelling_for_compression_s...	Input	☐	
Set_up_BOM	Input	☐	
Piston_Bowl_design	Input	☐	
Current_casting_Cam_design_15%_(incl...	Output	☐	
Lash-up_Twin_Turbo_arrangement_(incl...	Input	☐	

Fig. 6.6 Software interface to improve model building

Gantt chart models of design processes only capture a single task outcome. The newly developed interfaces allow the user to create alternative outcomes which could correspond to a delay to the task or a failure to contribute the expected information. For example, a testing task could be defined as having the following three outcomes: 1) success which leads to an increase in an associated parameter confidence; 2) failure which results in a decrease in confidence and 3) an inconclusive result such that confidence is unchanged. The modelling interface provides support in defining such alternative task mappings (Fig. 6.7).

Mapping for OutputState 1

Task Name: task_9

Input Output Resource

Buttons: Add, Delete

OutputState1 OutputState2

Cost Mean: 22.0 Duration Mean: 18.2

Cost Variance: 0.0 Duration Vari...: 0.4

Probability: 0.7

Parameter	Value
Parameter 1	10
Parameter 3	14
Parameter 6	16

Buttons: OK, Cancel

Mapping for OutputState 2

Task Name: task_9

Input Output Resource

Buttons: Add, Delete

OutputState1 OutputState2

Cost Mean: 12.0 Duration Mean: 38.2

Cost Variance: 0.0 Duration Vari...: 0.2

Probability: 0.3

Parameter	Value
Parameter 1	0
Parameter 3	1
Parameter 6	4

Buttons: OK, Cancel

Fig. 6.7 A signposting task with multiple outcomes

6.4 Input-model visualisation and analysis

Several new process-visualisations were implemented in the software tool to facilitate model-building and simulation-input model analysis.

6.4.1 Gantt chart and hierarchical Gantt chart

Because designers and managers at Perkins were already familiar with MS Project, and in order to satisfy requirement M4 (Section 5.1.5) – the need for conventional task-based process representations in Signposting – it was decided to create a Gantt chart representation of the model within the Signposting tool. Due to the size of the plans observed in industry, it was also necessary to implement a hierarchical Gantt chart (Fig. 6.8). These representations were a step forward from previous Signposting implementations which had limited capabilities for task-connectivity visualisation. As the Signposting model is very rich in data, a Gantt chart fails to adequately represent all of the information associated with the model. For example, alternative process routes cannot be shown on a single Gantt chart – however, it is possible to show the route through the process which corresponds to no failures – the route which is conventionally shown by Gantt charts in Perkins. Several further visualisations were developed to display other information captured by Signposting models.

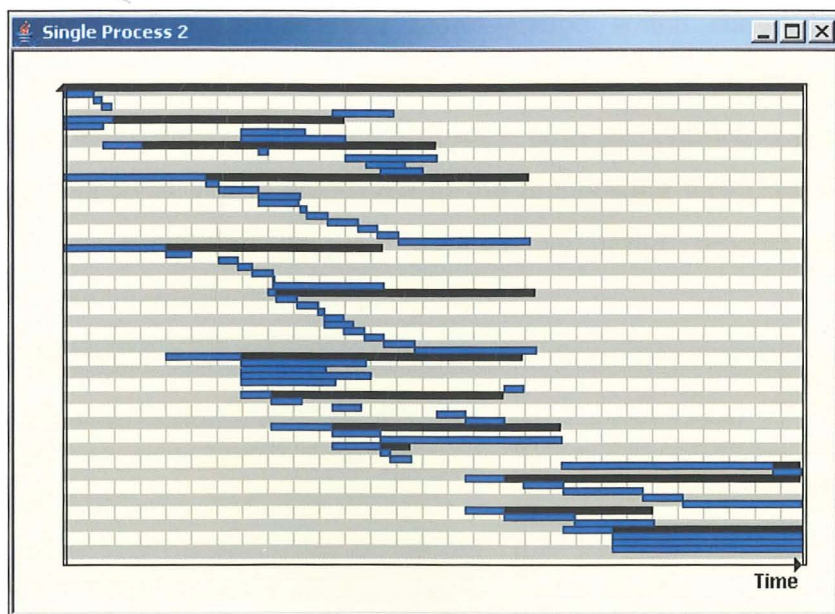
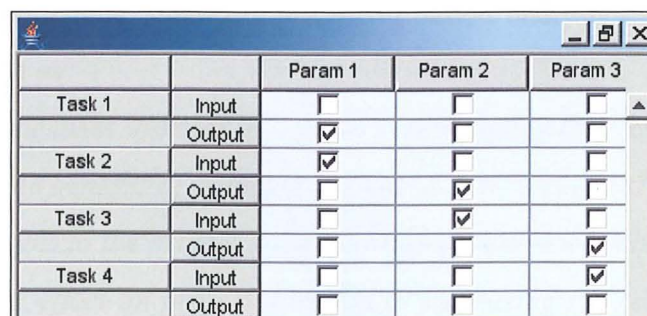


Fig. 6.8 Hierarchical Gantt chart (the black bars are task hierarchies)

6.4.2 Design Structure Matrix and Domain Mapping Matrix

In line with requirement M5 (Section 5.1.5) – the need for process visualisations to represent the connectivity of Signposting models – a Design Structure Matrix (DSM) was employed to give a better understanding of task dependencies and design iteration. This was complemented by Domain Mapping Matrices (Danilovic and Browning, 2004) containing tasks – parameter mappings, which can show how different tasks are connected through parameters (Fig. 6.9). While these representations were helpful, they failed to represent model data concerning parameter confidence. A novel representation of design data, parameter confidence profiles, was developed to address this issue.



		Param 1	Param 2	Param 3
Task 1	Input	☐	☐	☐
	Output	☒	☐	☐
Task 2	Input	☒	☐	☐
	Output	☐	☒	☐
Task 3	Input	☐	☒	☐
	Output	☐	☐	☒
Task 4	Input	☐	☐	☒
	Output	☐	☐	☐

Fig. 6.9 Task-Parameter DMM

6.4.3 Confidence profiles

Chapter 5 identified the need for a representation which shows how product confidence grows as the project progresses (Requirement C2, section 3.4.4). In order to satisfy this requirement, confidence profiles – plots of confidence growth against time – were conceived and implemented in the Signposting tool (Fig. 6.10). These plots show how parameters progress through the design process and give an impression of the interdependence of parameters.

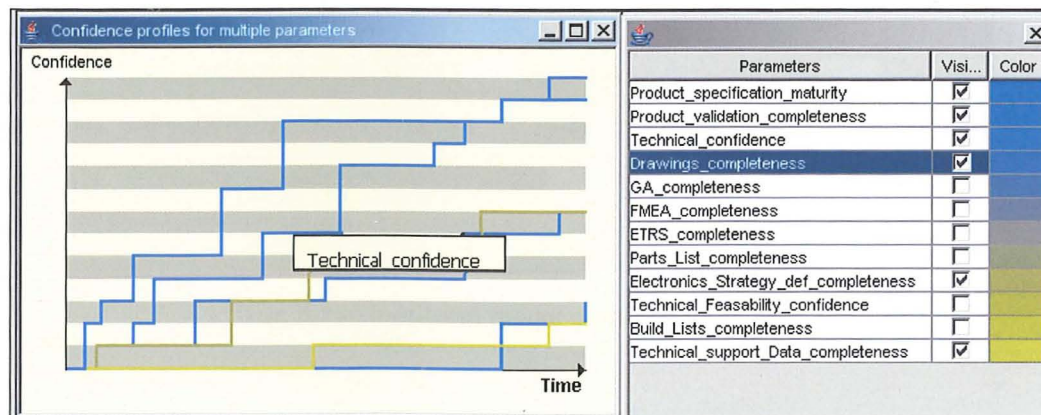


Fig. 6.10 Setting the visibility of different confidence profiles

Different colours in Fig. 6.10 represent different parameters and the parameters can be toggled between visible and invisible by checking the corresponding tick box. This allows the user to focus on different sub-sets of confidence profiles depending on the context of a particular query. The name of the parameter represented by a given plot can be determined by hovering the mouse over that trail.

6.5 Model perturbation capabilities

Once a model has been built, it can be perturbed to investigate sensitivity to different factors. Several different variations of the model can be created automatically. These different model variants can then be simulated and insights into high-level 'what-if' questions can be obtained. For example, perturbation analysis can provide insights into questions such as:

- *What if some tasks overrun by a given percentage of their expected durations?*
- *What are the benefits of reducing individual task durations?*
- *What happens to the project if the duration of task rework increases?*
- *What is the effect on project duration of increasing the resources by a given amount?*

A perturbation interface, which builds on appropriate extensions to the underlying software code, has been added to the Signposting tool (Fig. 6.11). These features allow systematic model perturbation and sensitivity analysis.

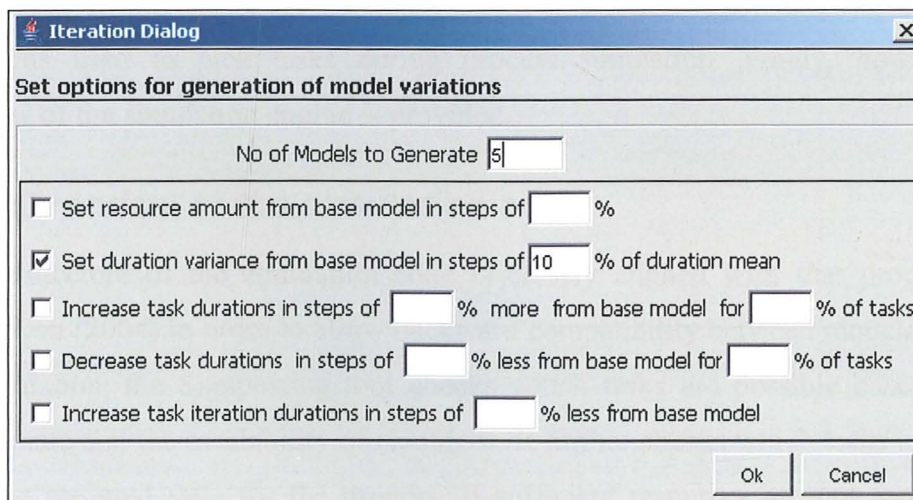


Fig. 6.11 High-level model perturbation interface

In addition to high-level perturbations, specific model perturbations are also possible using the dialogue box shown earlier in Fig. 6.6. Together, these different perturbation capabilities satisfy requirement T1 (Section 6.1.1) – the need for model-perturbation capabilities which facilitate sensitivity analysis.

Managers, who are concerned about a specific process risk such as a delay to a given task, can edit the model and simulate to determine how this change is likely to impact downstream process tasks. Duration uncertainty or cost uncertainty may also be modified, as can the likelihood of achieving a given outcome from the task. By making appropriate changes to the input-model, project sensitivity to risks can be explored. While valuable analyses can be performed by examining visualisations of different perturbations of the input-model, the use of simulation allows more thorough process exploration.

6.6 Enhanced simulation capabilities

Together with capabilities for improved model building and perturbation, simulation can provide useful insights into design process behaviour. Simulation allows assessment of model-response to different levels and types of uncertainties as well as different combinations of risks. Results from such assessments can be used to develop strategies for risk mitigation, such as robust task-ordering to reduce the probability of schedule overruns. This section describes changes to the Signposting simulation tool which focus on increasing the efficiency of the simulation engine and improving the algorithms used to pick tasks during process simulation. Firstly, however, an overview of the simulation engine is provided.

6.6.1 Overview of the simulation code

The architecture of the simulation code is closely aligned with that proposed by O'Donovan (2004) in order to allow backward compatibility between models. During the simulation, the Signposting tool checks which tasks are possible based on the current state and the availability of resources. It further checks which tasks are useful based on the goal state for the process. If sufficient resources are not available to execute all of the useful, possible tasks, a choice must be made about which task to do next. O'Donovan's code chooses tasks based on the input state (alternative task-picking strategies are proposed below in section 6.6.3). In many cases, more than one

task can be executed in parallel. The code checks whether more tasks can be started and, if so, executes these tasks. As the different tasks complete, the endTask class updates the process histories (updates different parameter states, returns resources to the resources pool, tracks which tasks have been completed). The success or failure of a task is determined by comparing the task's output state against either the current state of the process (or the task's input state depending on the user's preference). If a task fails, the process is reset to the highest valid previously-reached state. Tasks which were performed based on input states which are invalidated by the task failure are done again. If the task succeeds, the code checks whether the goal-state has been reached, in which case ongoing tasks are stopped. If not, the pickTask class is called again in light of the changed design state (Fig. 6.12). An example to demonstrate the execution of the simulation code is provided in the Appendix.

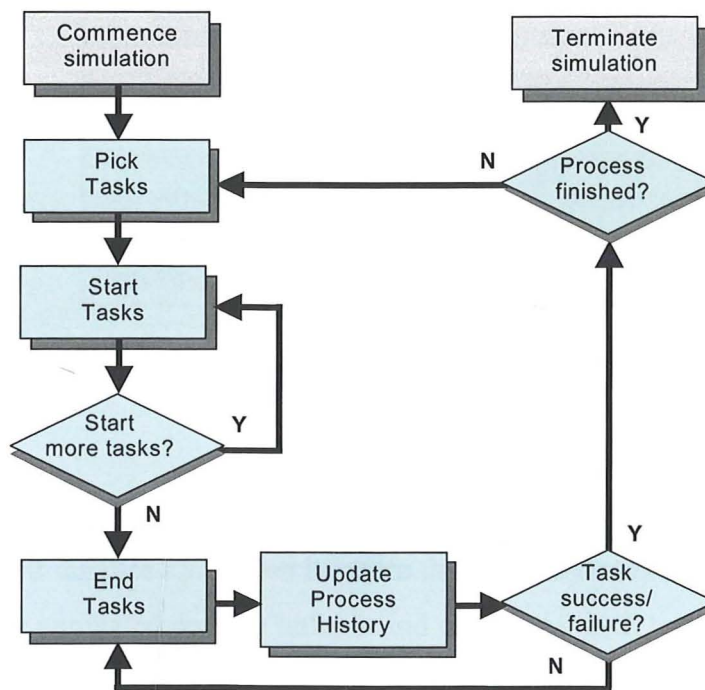


Fig. 6.12 Flow diagram of the simulation code

6.6.2 Enhanced simulation capabilities

In response to requirement T2 (Section 6.1.1) – the need to improve the execution speed of the simulation code, the efficiency of the simulation code was increased considerably by reconfiguring the data structures and changing the way in which key model variables were accessed and written to. As a result, the time taken to perform different simulation analyses was reduced from days to minutes.

The advantage of using data-structures to store simulation data is not limited to efficiency gains – it also allows the exploration of process properties which could not previously be simulated. An example of this is the simulation of major failures which require the rework of several tasks. In Signposting, failure causes the parameter confidence associated with a parameter to fall. If the parameter confidence at the output state is lower than the confidence at input, then the simulation code can search back through the process history until a valid state (higher than or equal to the failed state) is found and the simulation can recommence from the valid state forward (Fig. 6.13). The previous Signposting code (O'Donovan, 2004) did not have the capability to back-track and identify tasks performed based on incorrect input data (invalid assumptions about confidence).

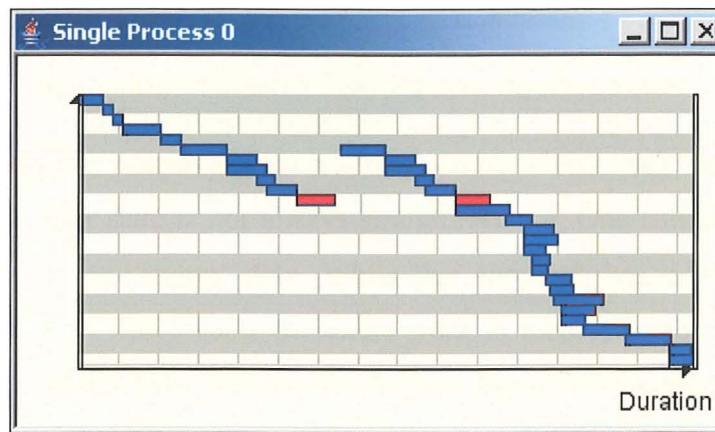


Fig. 6.13 Backtracking in the event of task-failure

Task-acceleration may also take place during iteration – when tasks are done for the second time, their durations are often likely to decrease. By tracking the events which take place as the simulated process unfolds and checking which tasks have been done before, the enhanced Signposting tool allows improved modelling of learning during iteration.

6.6.3 Improved algorithms for picking tasks

Requirement T3 (Section 6.1.1) defined the need to improve algorithms for picking tasks during the simulation of Signposting models. The previous implementation of the Signposting code picked the task with the highest input state. However, the highest input state does not guarantee the highest output and picking the task with the highest input state can result in better tasks being overlooked (Fig. 6.14 – Task A has a higher input confidence in parameter 0 but still achieves a lower output confidence).

This research implemented different algorithms for task selection, which reflect the manner in which designers look ahead at the different possible outputs which are likely to result from task-execution.

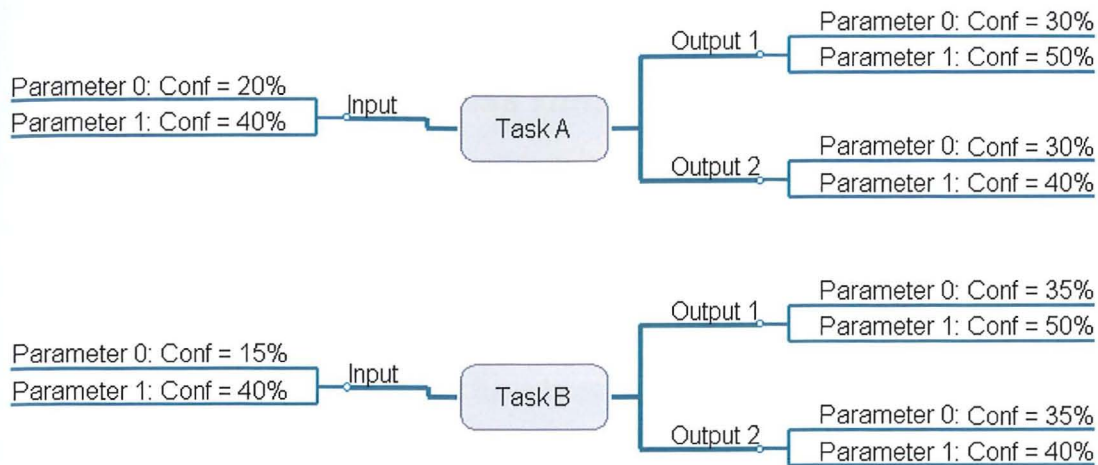


Fig. 6.14 The limitation of picking tasks based on input only

Signposting tasks can potentially have several output states such that picking a task based a single output state is not always straightforward. One solution is to pick the tasks based on the best output state. While this approach has limitations, it aligns closely to the way in which some designers work in reality, as observed in the case study. They choose the task that has the highest benefit if everything goes well. A pickTask algorithm which corresponds to choosing the best possible task outcome was implemented in the code.

However, many designers choose tasks based on the careful consideration of several possible outcomes, rather than just the single best outcome. This approach is captured in the extensions to the Signposting code through the use of task-choice based on an average output state. The averaged output state is calculated by weighting each output state for a given task based on the likelihood that of each outcome taking place. The most appropriate choice of pickTask algorithm depends on the model in question. If the tasks within the model have many output states, calculating average output states may decrease the efficiency of the code. Alternatively, choosing tasks based solely on the input state may result in the choice of sub-optimal routes through the process.

6.7 Output visualisation and analysis

Various new visualisations were implemented to facilitate analysis of the Signposting simulation data and to satisfy requirement T4 – the need for new visualisations of

model data and simulation results (Section 6.1.1). These included scatter plots, histograms, averaged Gantt charts and confidence profiles. The notion of “representative” process runs, as a means to summarise the results of several simulations, was also pursued.

6.7.1 Representative process runs

The simulation analysis produces very rich data – summaries such as representative process runs help overcome the associated interpretation challenges. Representative processes runs were identified by comparing the mean values of different process variables – duration, task-timing, parameter confidence, cost - observed during each simulation. Once these mean values have been calculated, a weighted average process is calculated and the simulation run, which most closely matches this process, is chosen as the representative process (the weightings of the different process variables are defined by the user). This process can then be shown in the form of a Gantt chart or DSM. A concern about this approach is that the average process may not be representative of a group of simulations – this may happen, for example, in the case of a bimodal distribution. Such cases can, however, be identified using other visualisations such as histograms, as discussed below.

6.7.2 Histograms and scatter plots

To alleviate such concerns about averaging effects, which can bias results, histograms and scatter plots for process cost and duration were developed (Fig. 6.15). These visualisations show which cost/durations are most likely for the project, as well as the ranges of possible values for both variables. In addition, the scatter plot is interactive: by double clicking on the dots in the scatter plot, the represented process simulation run is selected for more detailed analysis, for example using visualisations specific to the selected simulation instance (e.g. Gantt charts and DSMs as discussed below).

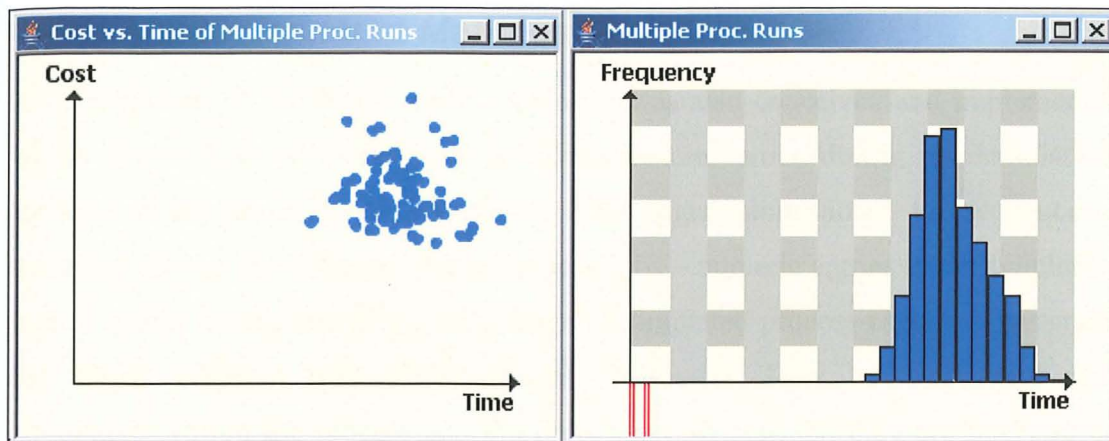


Fig. 6.15 Histograms and scatter plots show data from multiple process simulation runs

6.7.3 Gantt chart subject to uncertainty

A conventional Gantt chart does not capture any information about the uncertainty associated with task timing; conversely, it shows how the project will progress if everything goes right. This rarely happens in real design processes: as a result, the timing of tasks varies from that shown in the Gantt chart. Simulation analysis can predict the likely variation in task timing. An advanced Gantt chart was developed and implemented as part of this research – the shading indicates the earliest, latest and most likely timing for the different tasks within the process (Fig. 6.16). It maintains much of the simplicity of the conventional Gantt chart while nonetheless representing important information about process uncertainty.

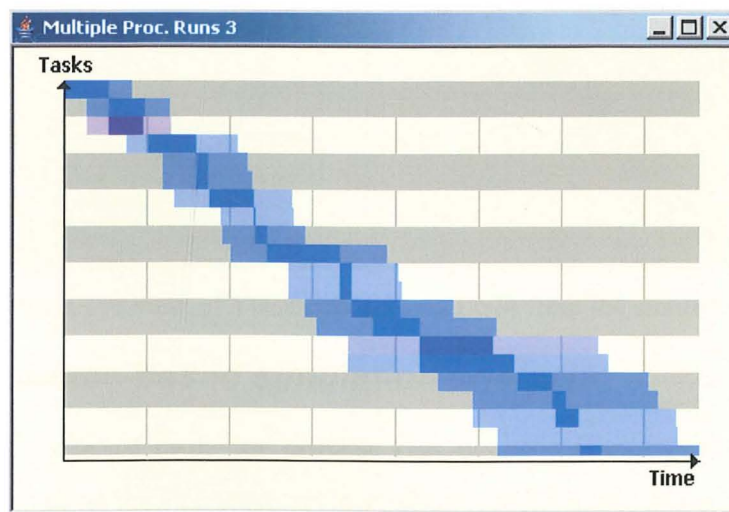


Fig. 6.16 An advanced Gantt chart for risky projects (the shading indicates the range of possible times during which a task is likely to be executed)

6.7.4 Design Structure Matrices

A development of the design structure matrix was also conceived and implemented which shows how often different task-dependencies arose during the simulation. Because of the flexibility of task-choice in the Signposting model, the sequence of tasks can change for different simulation scenarios – numeric entries in the simulation output matrix correspond to the percentage of simulated process-execution instances for which different task-sequences were followed. They thus highlight task alternatives within the process and the probability with which they are likely to be executed based on simulation (Fig. 6.17). The shading of the boxes indicates the strength of dependency between tasks (the darker the shade of blue, the stronger the link).

The Signposting software also allows the user to zoom in on matrix and use ‘fish eye’ views (Fisher et al., 1997) to see detail in context (Fig. 6.17). Designers can see how different tasks are connected at a local level while maintaining an appropriate perspective on the rest of the design.

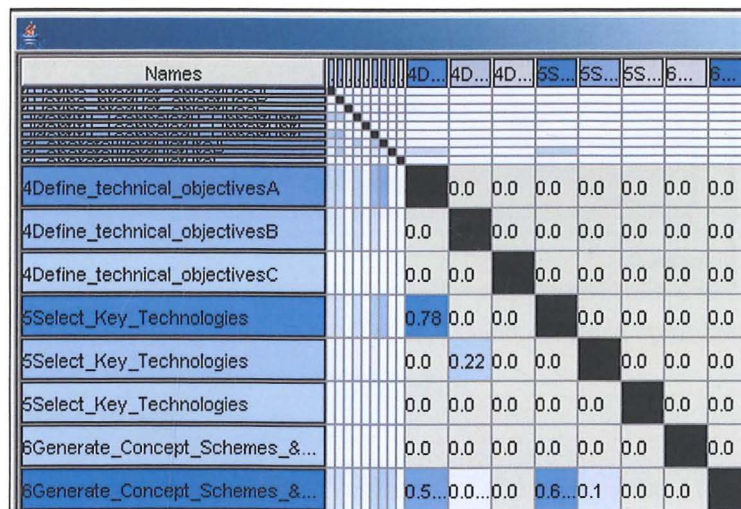


Fig. 6.17 Fish-eye view of a task-dependency DSM from the simulation output

6.7.5 Confidence-based simulation analysis

While existing software tools for process risk analysis (Pertmaster, Risky project, @risk as described in the literature review) show useful information about the trade-off between time, cost and resources, they are of limited value for design process analysis because they provide no insight into the manner in which these risks are likely to affect design confidence. The impact of various uncertainties on design project behaviour can be explored by simulating different task failures and delays and

investigating the effect on confidence profiles. In combination with Gantt charts, such profiles comprise an intuitive visual representation that can help designers and managers identify how different risks (e.g. task failures) are likely to affect parameter confidence values – it is possible to represent task failures (shown in red) in Gantt charts of the simulation output and determine the resulting effect to parameter confidence by examining the confidence profiles (Fig. 6.18).

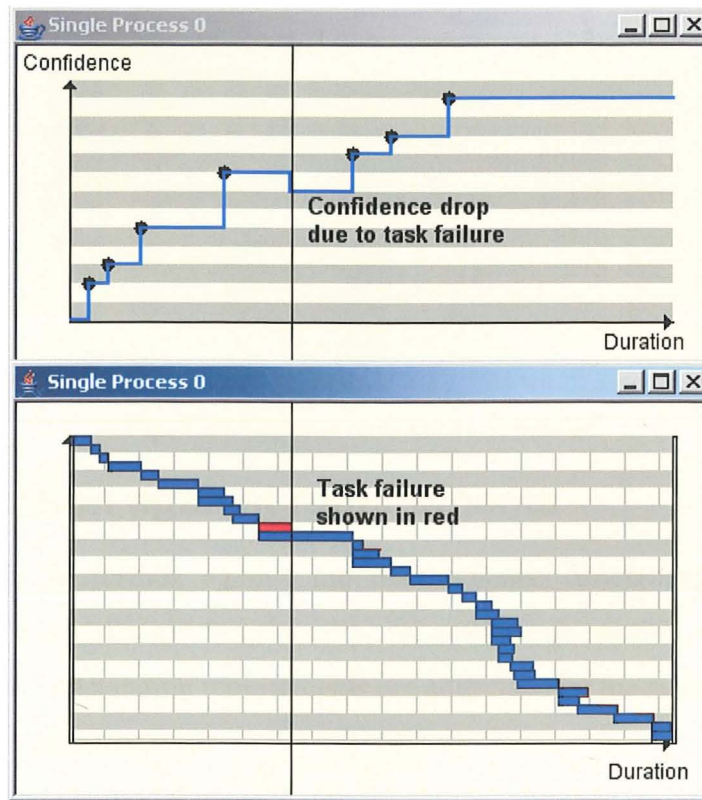


Fig. 6.18 Using confidence profiles and Gantt charts in combination

6.7.6 Multiple views

The Signposting tool provides alternative views of the same data – for example, Gantt charts and DSMs are both task-based representations, but both representations provide different information to the user, the former showing task timing while the latter conveys dependency information. Similarly, both scatter plots and histograms provide information about process duration likelihood, but the scatter plots also show how duration overruns impact project cost. The combination of visual representations provided by the Signposting tool (Fig. 6.19) overcomes the limitations of individual, isolated views, thus facilitating more extensive process analysis (Keller et al., 2006). More on the application of these new outputs will be described in the following chapters. Firstly, however the issue of verification is considered.

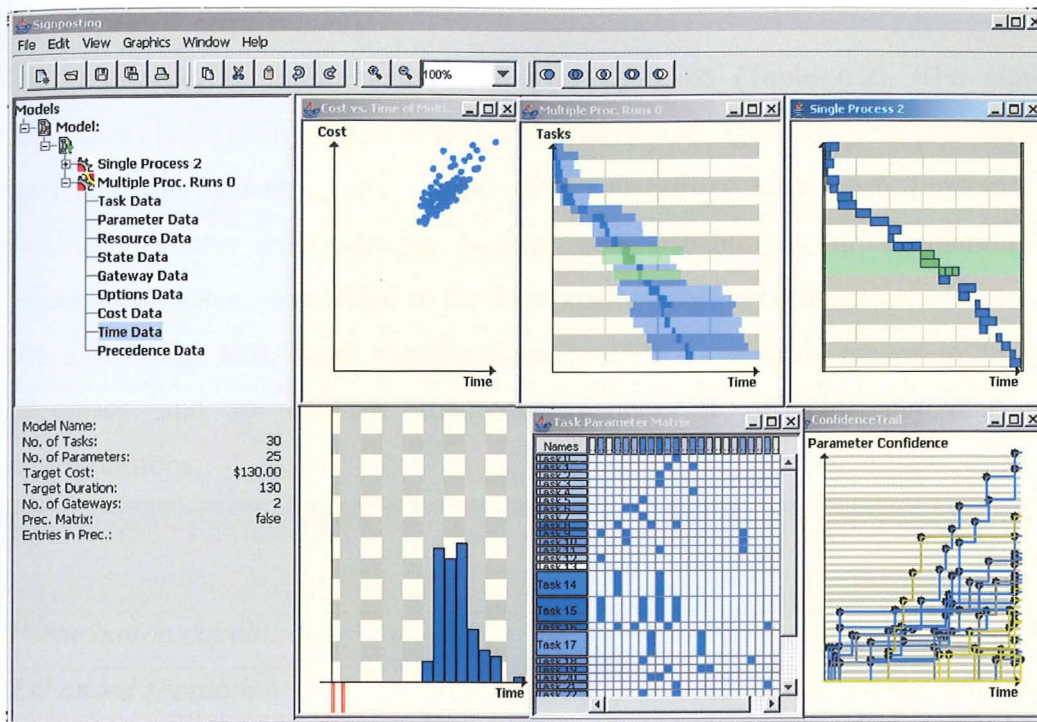


Fig. 6.19 Using several views in parallel

6.8 Verification against requirements

Verification was performed against the requirements, concerning modelling and representation, and software functionality, as outlined in section 6.1.1 and in the previous chapter. Outstanding requirements are addressed by the subsequent chapters.

Modelling and representation

The requirements for modelling and representation, outlined in section 5.1.5 were satisfied (Table 6.1). The changes to the model structure, which were also reflected in the code, allow the modelling of task precedences and parallel tasks which affected the same parameter. Several software features and new visualisations were implemented to ease the model building process.

Modelling and representation requirements	Requirement No.	Achieved	Section
Capability to model task precedences	M1	✓	6.3.1
Capability to model parallel tasks	M2	✓	6.3.2
Support for model building	M3	✓	6.3.3
Task-based process representations	M4	✓	6.4.1
Representations of model-connectivity	M5	✓	6.4.2

Table 6.1 Verification of requirements for modelling and representation

Tool-specific requirements

The tool-specific requirements were also achieved (Table 6.2). The simulation algorithms have been enhanced such that analysis of Signposting models can now be performed in real-time, and perturbation capabilities have been implemented to facilitate process interrogation. Several new capabilities for simulation output visualisation have been added to the Signposting tool. In conjunction with the rest of the simulation tool, these visualisations provide a valuable means to interrogate processes and to evaluate the consequences of different potential process configurations.

Tool-specific requirements	Requirement No.	Achieved	Section
<i>Perturbation capability</i>	<i>T1</i>	✓	6.5
<i>Enhanced simulation</i>	<i>T2</i>	✓	6.6.2
<i>Improved algorithms for picking tasks</i>	<i>T3</i>	✓	6.6.3
<i>Enhanced output model visualisation</i>	<i>T4</i>	✓	6.7

Table 6.2 Verification of requirements for modelling and representation

Other requirements

The analyses of hypothetical models was proposed as a means to explore structural variations of models without being constrained by process specifics (Requirement S1). The first two confidence-centred requirements were also achieved through extensions to the Signposting tool.

Other requirements	Requirement No.	Achieved	Section
<i>Structural variation analysis capability</i>	<i>S1</i>	✓	6.6
<i>Perkins compatible Signposting models</i>	<i>C1</i>	✓	6.3
<i>A representation of confidence growth</i>	<i>C2</i>	✓	6.7.5

Table 6.3 Verification of requirements for modelling and representation

The fulfilment of the above requirements results in a more practical Signposting simulation tool, which can provide better understanding of process-behaviour, thus facilitating the development of more robust plans.

6.9 Summary

The chapter described the conceptualisation and development of a tool for process modelling and analysis. The tool is inspired by previous research into Signposting but constitutes numerous enhancements in terms of support for model building, model perturbation, simulation analysis, and visualisation of both the input model and the simulation results. The chapter also introduced the notion of process analyses based on hypothetical models. Simulation analysis of both real-world and hypothetical models, using the improved software tool, is the core theme of the following chapters.

7

Lessons from simulation of hypothetical models

Previous chapters have highlighted the importance of scale and connectivity, rework and the product-process link in predicting process behaviour. This chapter explores these issues through the simulation analysis of hypothetical models, with the goal of defining heuristics for industrial project planning. In contrast to the subsequent chapter, which concerns project-specific simulation analysis of a real-world model, this chapter considers structural variations between models (see Chapter 5) for which real-world model-analysis is impractical, unsuitable and/or unnecessary (Fig. 7.1).

Generation of hypothetical models, which allow the analysis of structural variations, is the initial topic of the chapter. This is followed by descriptions of different analyses performed to explore structural variations between models. Validation of results against case study data, along with a discussion of the implications for industry, concludes the chapter.

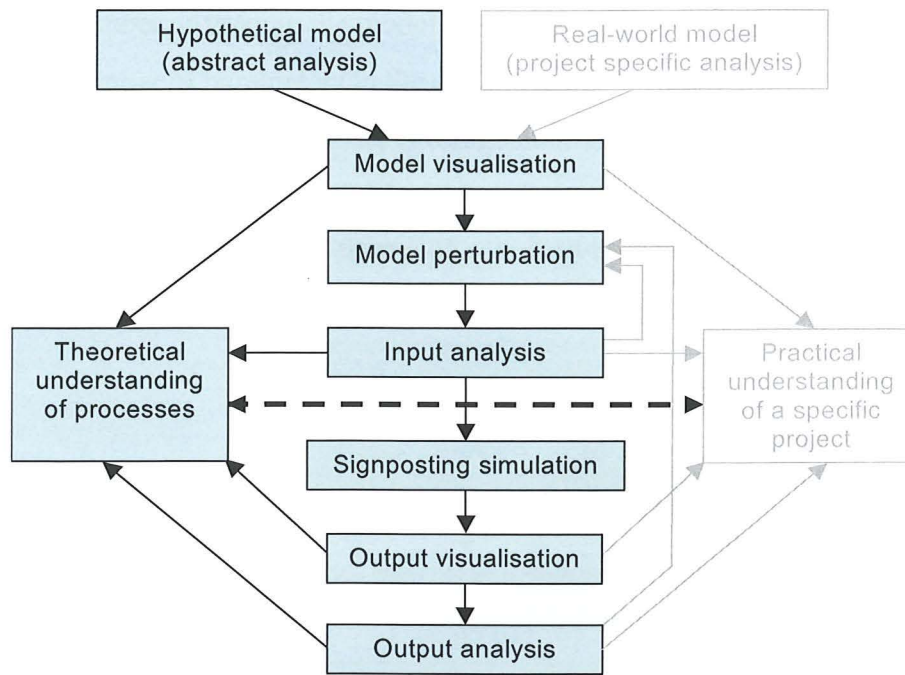


Fig. 7.1 Chapter focus with overall solution proposal

7.1 Generating hypothetical models

Chapter 6 discussed the theoretical advantages of process analyses based on hypothetical models. A means to create such models is crucial to the analyses described later in this chapter and hence to satisfying numerous objectives defined in chapter 5 (S2a, S2b, R1a, R2a, C3a, C4a). This section describes the development of a software tool for the generation of hypothetical design process models.

7.1.1 Generation of hypothetical models

The hypothetical models, which will be analysed in this chapter, contain the same elements as other Signposting models; the core differentiating factor being that they are not based directly on data from a real-world project. Nonetheless, they are made up of Signposting tasks, an initial state, a goal state and an assignment of resources. The Signposting tasks, in turn, specify input and output states in terms of parameter confidence values and precedences, and also contain data on costs, durations, resources and uncertainties (Section 3.4.3).

The steps involved in generating hypothetical models are shown in Fig. 7.2. Firstly, certain information must be provided. At a minimum, the following information is required:

- the number of tasks in the model,
- the number of parameters in the models and
- the level of task-connectivity (average number of connections per task).

Given this information, several blank task-instances are created and a DSM is constructed which defines the patterns of connectivity between tasks. The density for the DSM (number of connections) depends on the task connectivity-level specified.

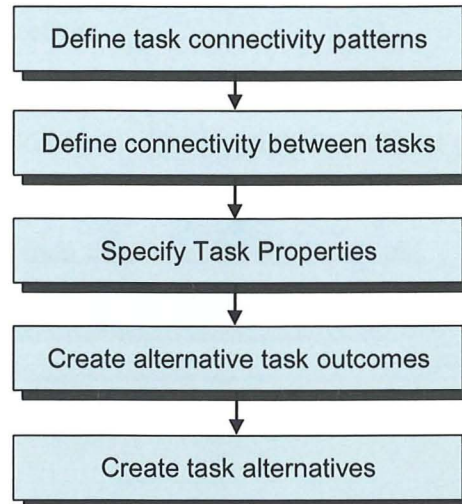


Fig. 7.2 Concept for generating hypothetical Signposting models

Next, parameters are assigned to the different connections – dependencies between tasks in Signposting models are always due to confidence or precedence parameters. The confidence contribution for each task is also assigned. The specific properties of the tasks, such as cost, duration and resource utilisation, are randomly determined but the hypothetical models can be edited to reflect different industrial contexts (Fig. 7.3). At this point, each task contains only a single outcome. Multiple outcomes are created by cloning the output state for a pre-defined percentage of tasks (the default is zero) and changing output characteristics, such as the confidence level, for different parameters or the task duration. Typically, multiple outcomes are defined only for a subset of the model's tasks. When multiple outcomes are created, the probability of different task outcomes occurring must also be specified. Task alternatives are produced by cloning specific tasks and modifying their characteristics to reflect changes in input information state. Once all of these steps have been carried out, the hypothetical Signposting model is complete, although further manipulation using the model building interface described in the previous chapter may be desirable depending on the specific goals of the researcher. An example hypothetical model is provided in the Appendix.

7.1.2 Implementation of the model generator

A software tool was implemented in Java™ to partially automate the model-building process (Java™ was chosen to ensure compatibility with the simulation tool). A model generation interface was developed to allow the specification of model characteristics, such as scale and connectivity level between tasks. Based on this information, the software creates a Signposting model following the approach outlined in the previous section.

Generated models can be visualised and edited using the Signposting, model-building interface (Fig. 7.3) – randomly assigned variables, such as costs and durations, can be manually overwritten, thus modifying the model to accommodate the user's intent. The resulting models can then be perturbed and analysed.

Generated Signposting Task

Task Name:

Input Output **Resource**

OutputState1 OutputState2

Cost Mean	<input type="text" value="7.0"/>	Duration Mean	<input type="text" value="14.3"/>
Cost Variance	<input type="text" value="0.0"/>	Duration Variance	<input type="text" value="0.4"/>
Probability	<input type="text" value="0.7"/>		

Parameter	Value
Parameter 5	<input type="text" value="1"/>
Parameter 7	<input type="text" value="3"/>
Parameter 14	<input type="text" value="2"/>

Fig. 7.3 A Signposting task-mapping from a generated model

Outputs from the Signposting analysis can be used to guide user inputs in creating further models in order to explore 'what-if' questions. Based on the analysis of several variations in model properties, insights into the project can be obtained (as described later in this chapter). Some useful insights were also obtained while developing the hypothetical-model generator, as discussed below.

7.1.3 Insights from the software development process

In addition to the potential advantages which can be gained from the analysis of hypothetical models, development of the model generation software proved a worthwhile exercise in its own right: it stimulated detailed consideration of the

abstract properties of tasks and their connections. In order to generate hypothetical process models, the software must define plausible task properties (in terms of task-connectivity, alternative outcomes etc.), an objective which requires in-depth consideration of the underlying properties of the process. In contrast to conventional model elicitation, which focuses on capturing existing links, model generation forces reflection on the range of possible connection-patterns between tasks. Thus, defining model properties for hypothetical models inspired abstract thinking about process characteristics which is not always required when creating more concrete models.

For example, modelling task-failure which leads to rework is difficult for hypothetical models. Even when considering real-world models where one can use historical data, predicting typical failures and their consequences is non-trivial; defining appropriate consequences for task failures in hypothetical models is especially challenging. Some failures require rework of only a few predecessors, others drive rework in several tasks. Some tasks are accelerated during rework, others not. Some task failures are discovered quickly, others go unnoticed for long periods of time. All of these issues were considered when creating the model generator. In addition to patterns associated with rework, patterns of task duration, duration uncertainty, task clusters and confidence growth were also considered.

Despite the insights gained from consideration of the possible space of design process models and the patterns associated with their constituent elements, a flexible, consistent set of rules for defining rework characteristics was not established. As a result, rework characteristics of the hypothetical models were assigned randomly and the user was empowered to edit the resulting models.

Nonetheless, the resulting thoughts motivated re-examination of the process modelling literature in a different light. This clarified the significance of issues such as rework discovery time, as discussed by Cooper (1993), and the concept of learning during iteration, as considered by Cho and Eppinger (2005). In addition, it underscored the lack of sufficient literature concerning the effect of process structure on plannability and highlighted the importance of simulation model verification.

7.1.4 Limitations of hypothetical, generated models

Broadly speaking, limitations of the hypothetical, generated models can be classified into two categories: 1) limitations of the approach and 2) limitations of the implementation.

One limitation of analysis using hypothetical models is that they do not correspond directly to real processes, and hence it is difficult to validate the results obtained from such models. Nonetheless, some evaluation of results is possible, especially in terms of more general insights obtained from such models (see Section 7.5).

In addition to concept-level limitations, implementation-level issues also limit the utility of the hypothetical models. The software assumes equally random durations and costs for tasks and assigns connections between tasks randomly (although the user can overwrite these random values). In reality, however, these process properties are not truly random, a factor which may limit the validity of the models.

7.2 Variations due to scale and connectivity

The case study and theoretical foundations chapters of this thesis showed that planning complex projects is difficult because sensitivity to process structure is poorly understood. Currently, large-scale design projects are being accelerated, task connectivity-level is increasing, and resulting process risks are unknown or unclear (Chapter 4). This section uses simulation analysis to explore the effects of different scale (number of tasks) and connectivity-level (number of task dependencies) on project performance.

7.2.1 Process model variations due to scale

While analyses of small models can provide many useful insights, such analyses ignore the impact of scale. In order to explore sensitivity to scale (requirement S2a, section 5.2.3) the scale of a hypothetical model was varied from 20 to 200 tasks in steps of 20 tasks. This range of values was chosen because it was large enough to highlight variations due to scale while small enough to simulate quickly using the Signposting tool. All models contained feedback loops – 10% of tasks had 2 possible outcomes: 90% probability of success which allowed the project to progress and 10% failure probability which led to rework of predecessors (Table 7.1).

Scale	Step size	No. of risky tasks	Number of simulations
20-200 tasks	20 tasks	10%	2000

Table 7.1 Project sensitivity to scale

Simulation analysis of the models was used to investigate the effect of scale on process duration. Results from 2000 simulations show that the variation in process-duration grows with increasing model size (Table 7.2, Fig. 7.4).

Scale	Mean (days)	Standard Deviation (days)
20 tasks	88.8	1.2
40 tasks	155.1	10.3
60 tasks	251.6	23.5
80 tasks	383.3	48.2
100 tasks	473.3	79.8
120 tasks	522.1	131.1
140 tasks	669.9	192.5
160 tasks	765	221.9
180 tasks	812.6	250.1
200 tasks	946.1	287.5

Table 7.2 Project sensitivity to scale

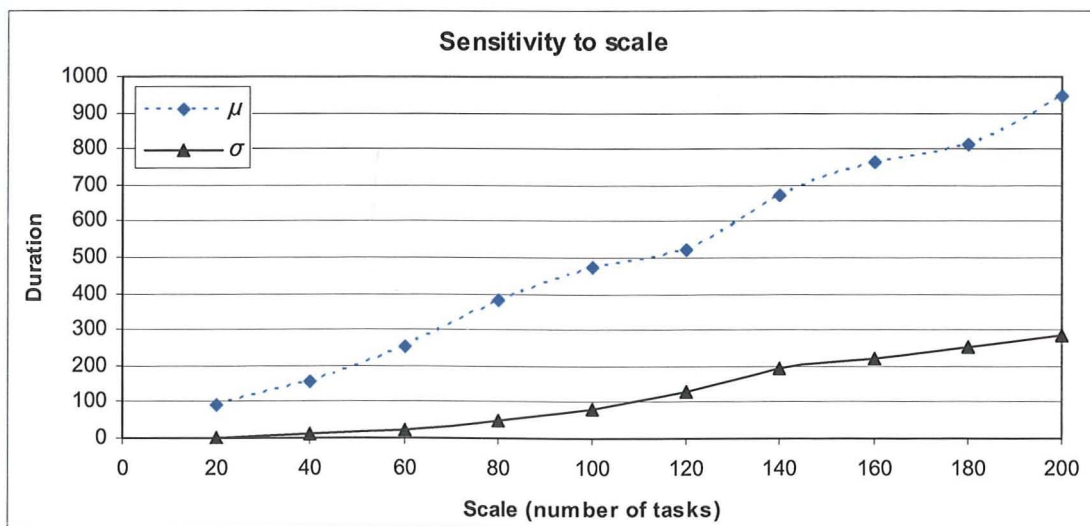


Fig. 7.4 Variation in project mean and standard deviation with scale

Examination of the simulated models, using the Signposting tool, shows that the number of process routes grows with model size due to the increased number of risky

tasks and task-failure modes. Even though the percentage of risky tasks remains constant, the actual number of tasks that can fail increases: in the small model there are only 2 tasks which can fail while the larger model has 20 possible sources of failure. Thus, the probability that some task fails in the small model is $1-(0.9)^2 = 19\%$ while the probability of at least one failure in the large model is $1-(0.9)^{20} = 88\%$. For the large model, it is highly likely that at least one task will fail driving process rework – multiple process failures are shown in Fig. 7.5. As a result, large models are more likely to exhibit greater variance in terms of the number of possible routes through the process (Fig. 7.6).

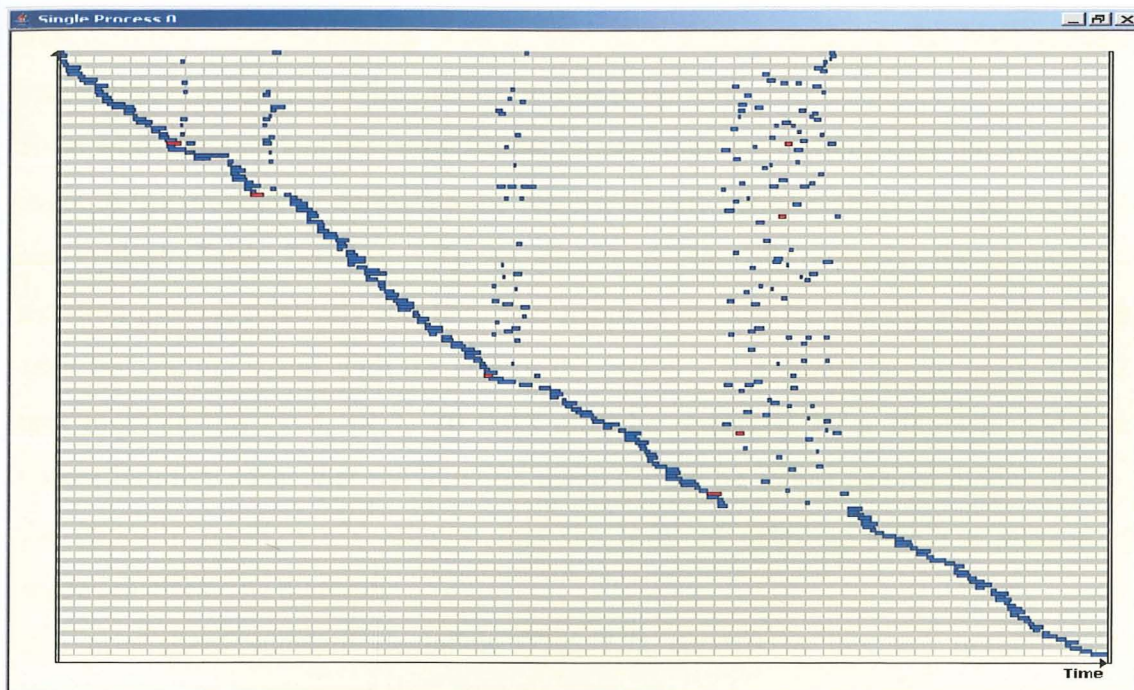


Fig. 7.5 Multiple failures in a 200-task process

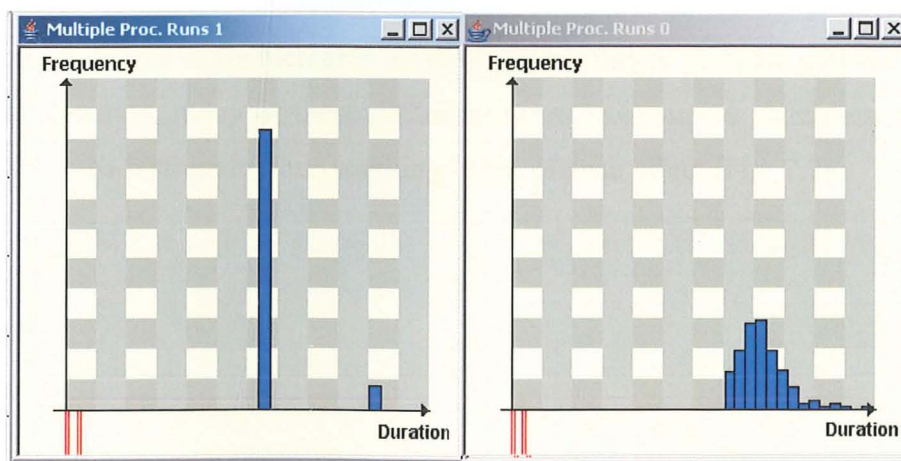


Fig. 7.6 Rising process variation with increasing scale – the 20-task process shown on the l.h.s. has only two possible duration outcomes while range of possible durations for the 200-task process (r.h.s.) varies considerably

The impact of the failure can be more varied for large models: failure in the smallest model considered above could, in theory, require the rework of up to 20 tasks; in the largest model, 200 tasks could require rework. Even if only failures which require

rework in 4 tasks are considered, there are $\binom{200}{4}$ (=64,684,950) combinations for the

larger model and $\binom{20}{4}$ (=4845) for the 20 task model. Although, it is likely that a

failure will only require rework of a subset of the maximum number of possible combinations, the calculations nonetheless illustrate how complexity can grow exponentially with increasing model size. This further explains the high process variability, highlighted by the simulation tool (Fig. 7.6).

Even when the standard deviation for the process duration is divided by the number of tasks in the process, to normalise the results, the effects of scale on project variance are evident (Table 7.3, Fig. 7.7). As shown in Fig. 7.7, however, the normalised duration-mean is relatively constant. This happens because the penalty associated with each task-failure scales with the size of the model – if a model has 20 tasks, then 2 task must be reworked, while 20 tasks must be reworked for a 200-task model. It makes intuitive sense that the duration-mean should be robust against model-scale – the opposite result would suggest that the level of detail to which a process is modelled affects its duration.

The normalised standard deviation (Fig. 7.7) appears to increase initially with increasing scale and then plateau. This happens because the small-scale models have a low number of failures associated with them and hence can only exhibit limited variation. As model-scale increases, so does the number of possible routes through the process. However, many of these different routes have similar durations and appear identical on the histogram, thus resulting in the plateau effect (Fig. 7.7).

Scale	Mean /Number of Tasks	Standard Deviation /Number of Tasks
20 tasks	4.44	0.06
40 tasks	3.88	0.26
60 tasks	4.19	0.39
80 tasks	4.79	0.60
100 tasks	4.73	0.80
120 tasks	4.35	1.09
140 tasks	4.79	1.38
160 tasks	4.78	1.39
180 tasks	5.51	1.39
200 tasks	4.73	1.43

Table 7.3 Project sensitivity to scale (normalised by number of tasks)

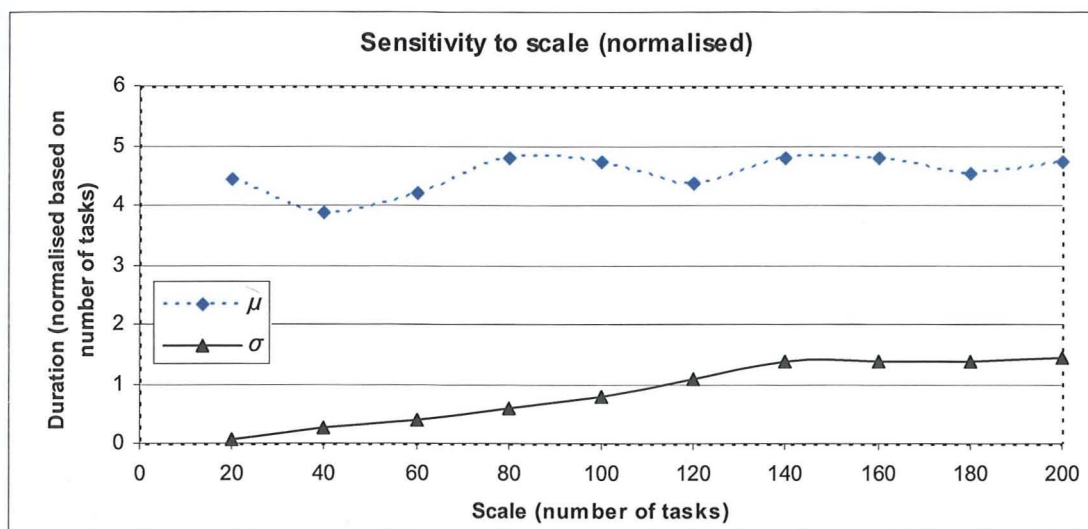


Fig. 7.7 Variation in percentage project duration uncertainty with scale (normalised by number of tasks)

Results also show that uncertainties accumulate and that planning the later tasks in a large plan is extremely difficult. Even if percentage delays at a project level remain similar, the actual uncertainty in timing of later tasks is massive for the larger projects.

It is important to note that the above analyses consider the scale of the model as opposed to the scale of the process, even though large-scale models can be used to define small-scale projects and vice-versa, depending on the chosen level of detail.

Thus, the results show that if a similar level of detail is used to model small- and large-scale projects, the latter are likely to exhibit greater levels of variation (in terms of possible task-order) and hence become more difficult to plan. Also, if the same project is planned at different levels of abstraction, the detailed plans give more specific information about the nature of task-failure with the downside of increased cognitive burden in terms of the larger range of possible failures that must be accounted for. For example, if the detail of a model is increased 10 fold, then a single failure in the abstract model can manifest itself as numerous combinations ($\binom{10}{n}$ for $n=1-10$) of failures in the detailed model. From this perspective, deciding on an appropriate level of detail is a trade-off between achieving sufficiently specific understanding of task-failures and avoiding excessive cognitive burdens and modelling challenges.

While the above analyses provide insights into process behaviour, they are nonetheless open to criticism. The models make assumptions about project properties such as connectivity-level, task-failure levels and duration uncertainty – results may not hold true if variations in these properties overshadow variations due to scale.

Despite these concerns, the results reported in this section highlight limitations of making inferences based on small, simple models of the design process. Using results from models of small-scale projects, and extrapolating the results to larger projects, can be dangerous: much of the challenge lies in thinking large-scale. Although simple models can be useful, as demonstrated in the subsequent section on rework (Section 7.3), it is important to understand their scope and acknowledge their limitations.

7.2.2 Process variation due to connectivity-level

This section explores the impact of changing connectivity level on process performance (requirement S2b, section 5.2.3). To this end, the connectivity-level between tasks (the filling grade for a DSM) was varied from 5% to 30% in steps of 5% and 1200 simulations were executed on a 100-task model. (Levels of connectivity greater than 30% were not analysed because such model-characteristics were not common in the literature of case-study process models).

Filling grade	Step size	Model size	Number of simulations
5 % - 30 %	5 %	100 tasks	1200

Table 7.4 Project sensitivity to connectivity-level

Overall, the results show that as the connectivity level rises, both the mean and the standard deviation for the process increase (Table 7.5, Fig. 7.8). The analysis showed that increasing the connectivity level restricted the number of tasks that could be executed concurrently (Fig. 7.9) and increased the impact of task-failures (a greater number of dependencies means that more tasks must be reworked in the event of failure). Both of these factors impact project duration, although the influence of concurrency-level declines as models reach a saturation point in terms of connectivity-level – as the connectivity level rises the task-order becomes increasingly constrained until a point is reached when the introduction of further dependencies has little or no effect. For example, if task C depends on task B and task B depends on task A, then there exists an indirect dependency between C and A and the addition of a direct dependency between these tasks is immaterial. Before reaching the saturation point, however, tasks that had previously been performed in parallel to slower, critical tasks are increasingly performed in series, increasing the total number of critical tasks. As a result, the process displays higher variance.

Connectivity level	Mean (days)	Standard Deviation (days)
5%	200.35	28.6
10%	231.93	35.31
15%	282.08	79.05
20%	369.77	89.25
25%	477.58	114.15
30%	516.38	140.54

Table 7.5 Sensitivity to connectivity-level

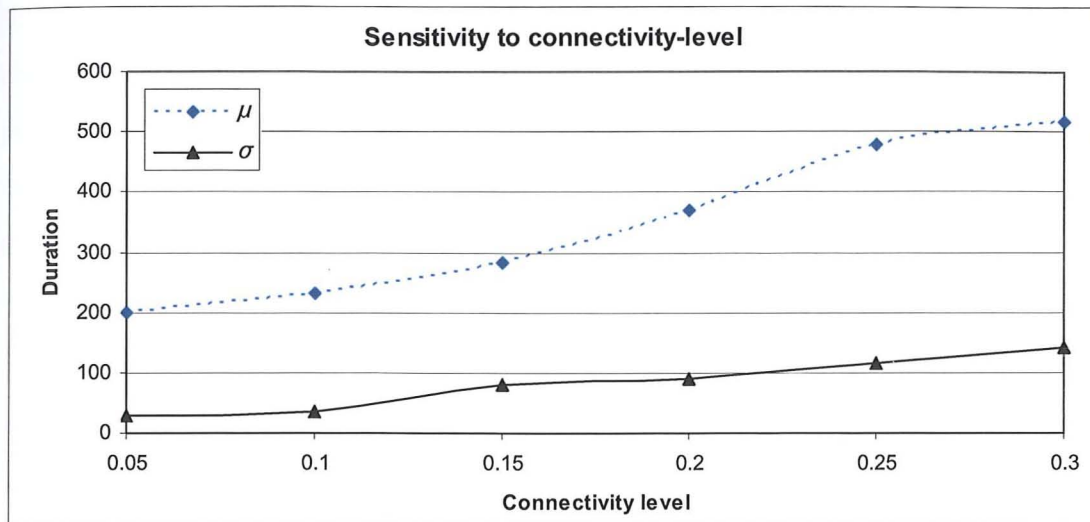


Fig. 7.8 Increasing duration mean and standard deviation with increased connectivity

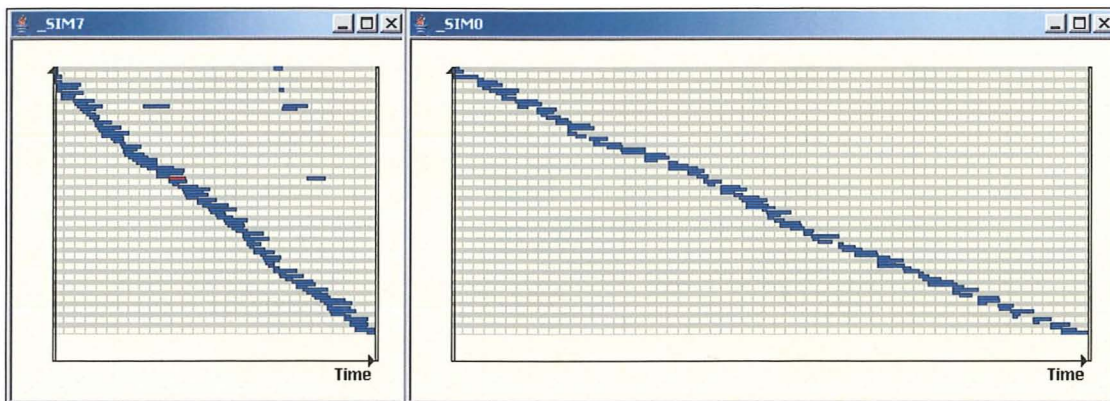


Fig. 7.9 Reduced concurrency due to increased connectivity. The higher degree of task-parallelism in the l.h.s. Gantt chart corresponds to a filling grade of 5% while the slower process on the r.h.s. corresponds to a filling grade of 30%

Simulation analyses were also performed to assess the variance of connectivity level with scale. To this end, the models described in section 7.2.1 were perturbed to increase the connectivity level by 50% (from 10% to 15%) and a further 2000 simulations were run. Results show that increasing connectivity has a more dramatic impact for the large-scale models but that the trends observed from the separate analysis of scale and connectivity are also present when both factors act in combination (Table 7.6, Fig. 7.10). As both scale and connectivity-level contribute to complexity, it is not surprising that the effects of changing the connectivity level are amplified for larger models. The results suggest that variation in large-scale projects could be decreased by reducing inter-task dependencies, a goal which can be accomplished through the use of staged gateways.

In the earlier section on complexity (Section 3.1), it was noted that number of connections between the elements reflects the complexity of a system. From this perspective, the use of gateways could be seen as a complexity-reducing mechanism in so far as it reduces process-connectivity. The simulation analyses show how process robustness varies with scale and connectivity-level and thus indicates that optimal gateway timing is likely to vary in different contexts.

Scale	Mean (days)	Standard Deviation (days)
20 tasks	102.3	1.2
40 tasks	206.9	16.5
60 tasks	337.6	30.1
80 tasks	428.7	38.3
100 tasks	585.6	51.2
120 tasks	764.7	173.2
140 tasks	935.3	249.2
160 tasks	1,059.7	332.6
180 tasks	1120.0	317.6
200 tasks	1207.4	388.2

Table 7.6 Sensitivity to scale and connectivity-level

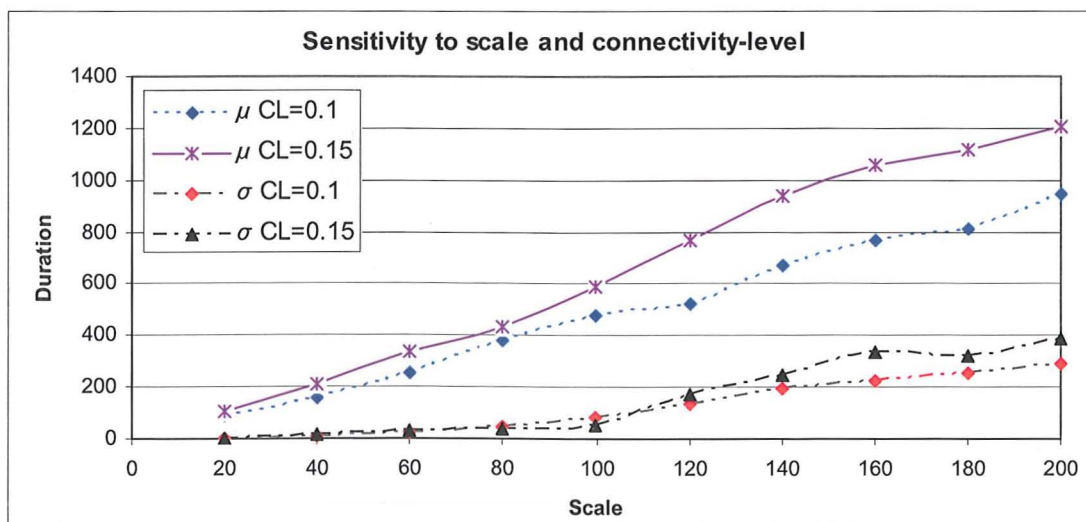


Fig. 7.10 Sensitivity for scale and connectivity-level

7.3 Variations due to rework

Process sensitivity to rework is poorly understood. During the case study, several different aspects of rework behaviour were observed; the duration of reworked tasks, the task-order during rework, the task rework-probability and the resulting reshuffling of downstream tasks all impact project duration. In contrast to manufacturing environments, where the scope of rework is often clear, rework of design tasks can involve the conception of a completely new approach which is comparatively costly and time consuming. Conversely, rework may be extremely quick – if a CAE test fails, rework may be contained to tweaking of a few key parameters in the software in order to achieve successful results. This section uses the Signposting tool to perform simulation of hypothetical models to analyse process sensitivity to task acceleration during rework.

7.3.1 Variations due to task acceleration during rework

Simulation analyses were performed to address the need to explore variations in task duration during rework at a structural level (requirement R2a; section 5.3.3). A simple 20-task model, created using the model generator, was used as the basis for these analyses. Three perturbations of the model were created to explore the manner in which different rework behaviours influenced process performance. For each of the three perturbed models, the duration of reworked tasks was modelled as 0% (zero-duration rework), 20% (accelerated rework) and 100% (slow rework) of the original task duration. These simulations were undertaken to determine the importance of accurately predicting rework duration, an issue which was noted as problematic during the case study chapter. Each model was simulated 1000 times.

Rework duration	Model size	Number of simulations
0%, 20% and 100% of original duration	20 tasks	3000

Table 7.7 Project sensitivity to rework-duration

The effects of different rework learning models (zero-duration rework, accelerated rework and slow rework) on project duration are shown in Fig. 7.11. All three histograms are based on the same model of task connectivity (15%) and resource availability: they differ only in the duration of reworked tasks.

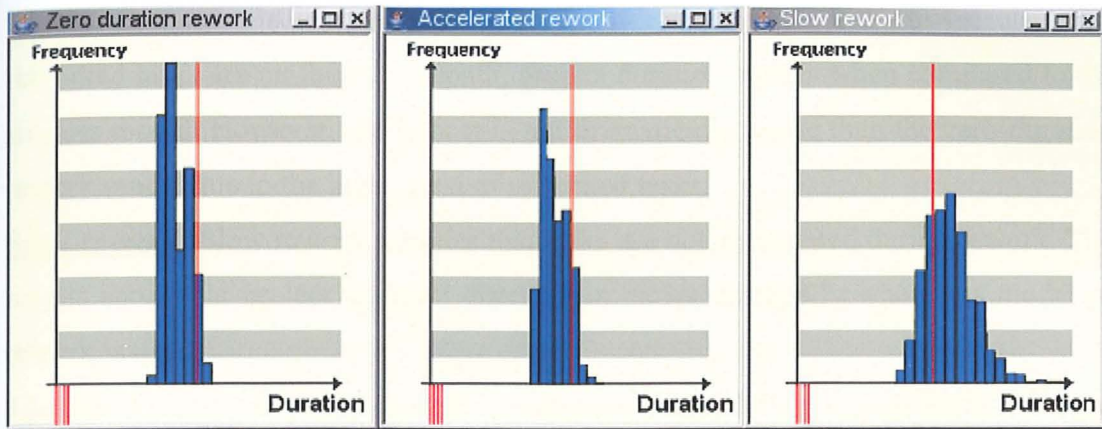


Fig. 7.11 Increasing the duration of task rework increases the project duration mean and variance

The results, based on 3000 simulation runs, show that the choice of rework model has a major influence on the expected project duration, even when deterministic (as opposed to probabilistic) task duration values are used. Although it makes intuitive sense that project duration will increase in response to longer rework times, the results highlight the importance of correctly predicting the type of iteration in design project planning. The target duration for the project, represented by the vertical red line, can be used as a reference point for comparing the different simulation results. The means and standard deviations for project duration are presented in Table 7.8.

	Mean (days)	Standard Deviation (days)
Zero-duration iteration	159.80	14.06
Accelerated iteration	167.61	15.99
Slow iteration	198.91	30.62

Table 7.8 Comparison of different models of learning during iteration

Zero-duration rework: Figure 7.11 shows that even when tasks can be reworked extremely quickly (instantaneously), considerable project delays are likely. While this model is an oversimplification of reality, it nonetheless underscores the importance of rework when estimating project duration: as task durations are modelled as deterministic, all project duration variability is due to rework. The reason for this increased duration is discussed at the end of this section.

Accelerated rework: The second rework model shown in Fig. 7.11 assumes that tasks take only a fraction of their original duration (20%). This model is appropriate for some design tasks where minor modifications are required based on information

which was not available at the time when the task was first performed. Because some reworked tasks are on the critical path, project duration suffers when compared to the simpler model. However, the impact is not dramatically worse than the zero-duration rework model due to the high speed of reworked tasks.

Slow rework: Slow rework assumes that tasks are not accelerated during rework. The results show that project duration distribution varies drastically when this model of rework is used.

The above results show that correct rework-duration modelling is important in design-project duration-prediction. This observation underscores a limitation of current project planning tools, such as MS Project™ and Primavera™, which largely ignore rework, and of DSMs (Browning, 2001) which offer little insight into the effects of different rework behaviours. This work shows that it is not sufficient to identify the existence of rework; the specific nature of rework must also be considered.

The analyses also satisfy the need to explore how undiscovered rework affects the downstream task order (requirement R3; section 5.3.3). The results show that even when the actual rework activities are performed extremely quickly, they can still have a major indirect influence on project duration by modifying the task order. This happens because the lower confidence state, resulting from failed and partially failed tasks, limits the choice of available tasks and ultimately delays the project. An example of this situation would be when a subcontractor provides insufficient or inaccurate data about a component design; inaccurate data leads to rework of dependent tasks while insufficient data delays their execution (Fig. 7.12).

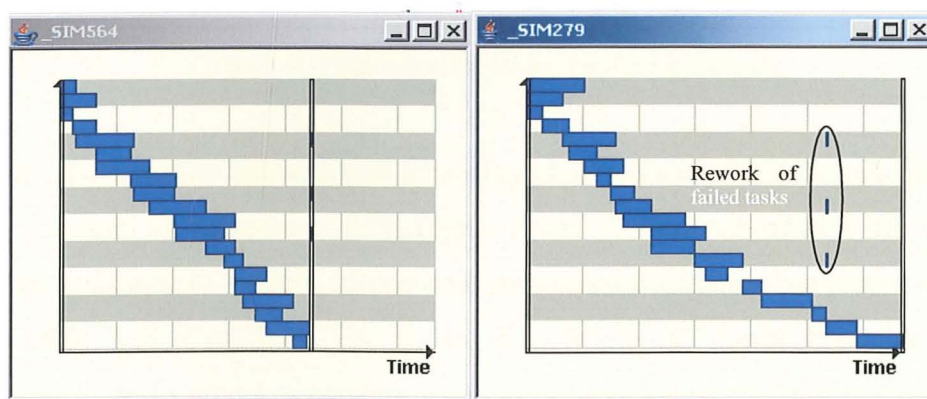


Fig. 7.12 The Gantt chart on the l.h.s. of the figure corresponds to a simulation with no failures. The Gantt chart on the r.h.s. corresponds to a simulation run with 3 failed tasks which lead to zero-duration rework. Even though these tasks are reworked instantaneously, the changed task-order due to their failure reduces task-parallelism and delays the project.

Results show that even when the duration of reworked tasks is short, major project delays are possible. This happens because designers are forced to press on with nominally downstream activities, based on preliminary information. These downstream activities, in turn, may require rework if the information on which they are based turns out to be incorrect. Likewise, these activities may also produce incorrect information leading to rework of other dependent tasks. Through this mechanism, rework can lead to a vicious circle of further rework and project delays as described by Cooper (1993). This is more likely to happen for projects that exhibit a high degree of connectivity, especially in the context of late discovery of rework, while projects that contain a high number of independent tasks usually result in more localised rework.

7.3.2 Sensitivity to rework-likelihood values

A similar 20-task model was also used to investigate sensitivity to rework likelihood (requirement R1a, Section 5.3.3). Six perturbations of the model were created. In each of these, two tasks were modelled as having two possible outcomes: one successful and one corresponding to a failure which required rework of predecessors. The task-failure probability for these tasks was varied from 5% to 30%. Each model was simulated 500 times in order to determine the impact of rework likelihood on project duration.

Failure probability	Step size	Model size	Number of simulations
5 % - 30 %	5 %	20 tasks	3000

Table 7.9 Project sensitivity to rework-likelihood

The simulation results showed that increasing the rework probability increased the project duration: an increase of 25% (from 5% to 30%) in the rework likelihood of two tasks translated to a 226% increase in the standard deviation for project-duration mean (Table 7.10, Fig. 7.13, Fig. 7.14). This result has implications for process modelling: in cases where accurately estimating task-failure probabilities during the project planning phase is difficult, erroneous project-duration-variance estimates are likely to result. Hence, considerable effort is warranted in ensuring that task-failure probabilities are as low as possible and that correct estimates for rework probabilities are used in models.

Rework likelihood	Mean (days)	Standard Deviation (days)
5%	146.53	7.69
10%	150.35	12.88
15%	154.11	16.05
20%	155.93	16.46
25%	160	19.47
30%	165.18	25.09

Table 7.10 Sensitivity to rework probability

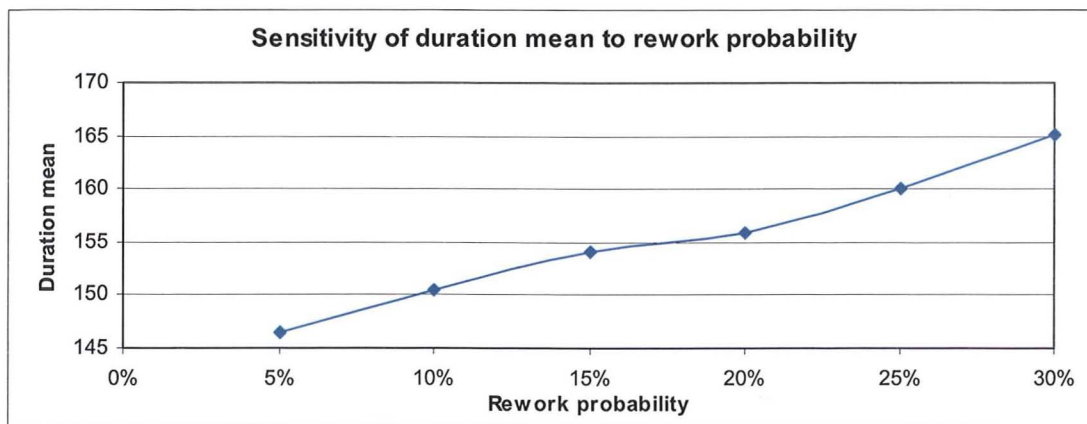


Fig. 7.13 The impact of rework probability on duration mean

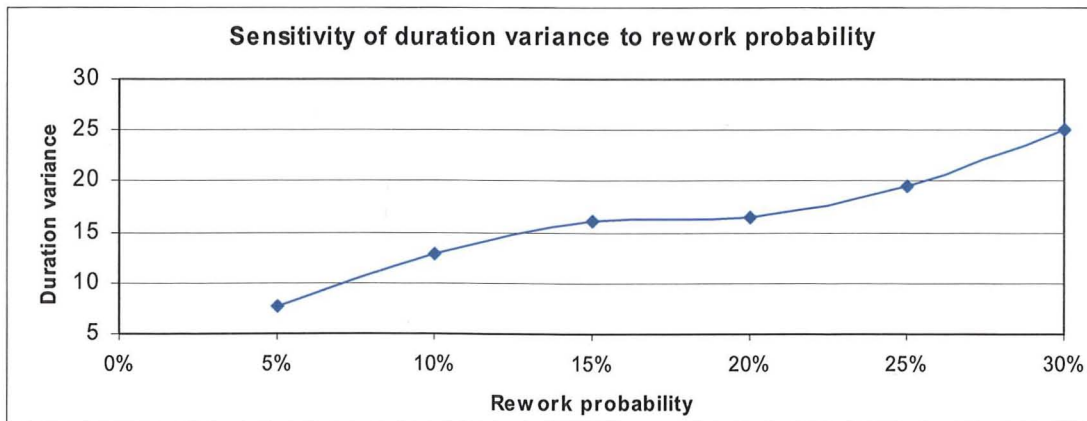


Fig. 7.14 The impact of rework likelihood on duration variance

Looking more closely at the simulation results, it can be seen that the impact on the process varies with different task failures (Fig. 7.15). Failure of task 9 only requires rework of 3 other tasks while the penalty when task 10 fails is rework of 5 tasks.

When both tasks fail in combination, the impact on the project-duration varies depending on the order of the task-failures (whether 10 failed before or after 9) due to task failure interaction (Section 5.3.1). In such cases the resulting process delay cannot be predicted by simply summing the delays due to the individual failures. Different failures can lead to rework in the same tasks – hence if two failures happen together, the resulting rework required will be less than that obtained by summing the rework required by individual failures. Also, the implications of failures may vary in light of upstream process variations which result in a changed design context.

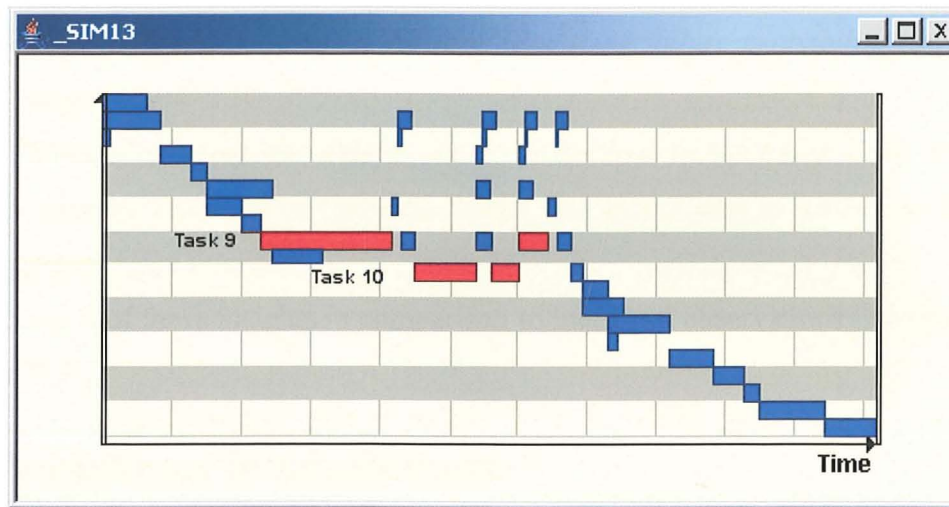


Fig. 7.15 Interaction of task-failures (appropriate task-ordering can reduce the impact of multiple task-failures)

It would be useful to perform more extensive simulation analyses to explore further combinations of task-failure probabilities (e.g. one task has a 10% chance of failure while another has only 5%). Also, it would be interesting to vary the rework discovery time in parallel to varying the rework probability. Such analyses were not undertaken because the set of possible combinations for hypothetical models is infinite. Instead a subset of these issues is considered for a specific real-world model (Chapter 8).

7.3.3 *Insights into rework*

The above analyses provide the following insights into rework and hence indirectly into the design of more robust plans:

- 1) *It is not always the time spent reworking tasks that causes delays – resulting changes to the task order can delay the project.*
- 2) *Task-duration during rework has a major impact on project duration. However, the benefits of task acceleration during rework can be overshadowed by the required changes to the task-order.*
- 3) *Getting accurate estimates for rework probabilities is vital in order to accurately predict project duration. Reducing failure probability can have a major effect.*
- 4) *Simply summing the delays due to individual task-failures does not yield accurate estimates for process delays, due to the way in which the different failures can interact.*

A discussion of these insights in comparison to industrial observations is contained in section 7.5.2.

7.4 Variations in confidence

Chapter 5 described how confidence data could be applied to process-analysis. At Perkins, products are designed subject to tight deadlines. While traditional analyses have focused on the likely variations in project durations, Perkins is interested in exploring how product-confidence is likely to vary subject to a given duration.

7.4.1 *Variations due to the timing of confidence-growth*

Simulation analyses were undertaken to explore process sensitivity to the timing of confidence growth in the context of rework (requirement C4a; section 5.4.4). To this end, three 20-task models were created, to reflect high confidence-growth at different stages of the project (Table 7.11). In the first model, 60% of the confidence was expected upon completion of the first 7 tasks; in the second model 60% confidence was not achieved until completion of the 14th task; in the final scenario, 60% confidence was expected on executing the 17th task, the remaining 40% anticipated right at the end of the project. 10% of the tasks acted as rework drivers and the model

did not account for uncertainty in task-duration. The duration of reworked tasks was 20% of their original duration. Each model was simulated 1000 times.

Timing of confidence growth	Model size	Number of simulations
Early, middle, late	100 tasks	3000

Table 7.11 Project sensitivity to timing of confidence-growth

Results showed that when much confidence growth is expected late in the process, high variations in confidence are likely if tight deadlines are imposed (Table 7.12, Fig. 7.16).

	Mean confidence at deadline	Standard deviation in confidence at deadline
Early confidence growth	95.33	5.04
Even confidence growth	90.73	12.75
Late confidence growth	85.28	17.23

Table 7.12 Comparison of different models of learning during iteration

Examination of the simulated data shows that task-failures are particularly hazardous in the context of late confidence growth because sufficient time is not available to rework failed tasks. If the majority of the confidence is gained early in the process, the impact of these task-failures on confidence is diminished – the process has time to recover – and reasonably high confidence can be expected even if some tasks fail. Thus, the shape of the different confidence profiles provides clues as to which components are at the highest risk: components that mature late in the design process are less forgiving if problems occur.

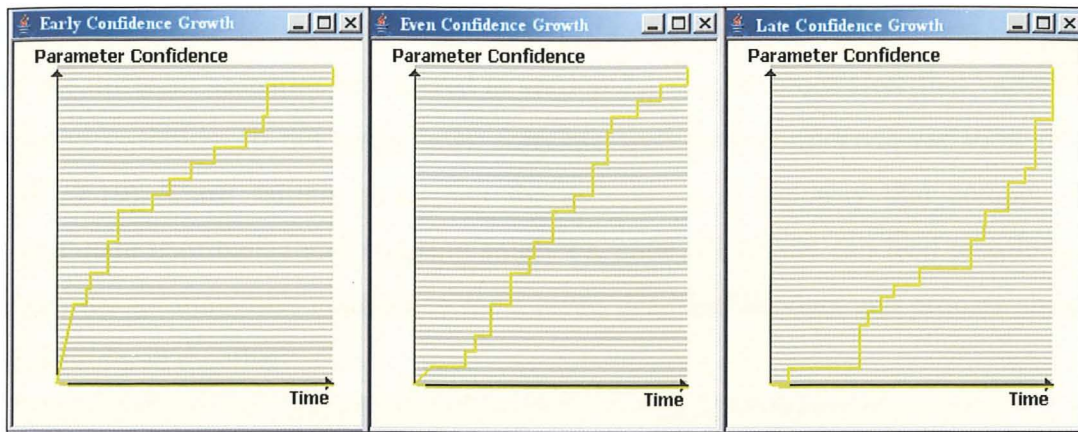


Fig. 7.16 Typical confidence profiles for each scenario

7.4.2 Confidence variations subject to tight deadlines

Chapter 5 defined the need to simultaneously examine processes from a task-based and a confidence-based perspective (requirement C3a, section 5.4.4). Confidence profiles show that processes that look similar in a task view can be very different when looked at from a confidence perspective (Fig. 7.17). These profiles often have different shapes because the same activities contribute differently to the confidence associated with different components.

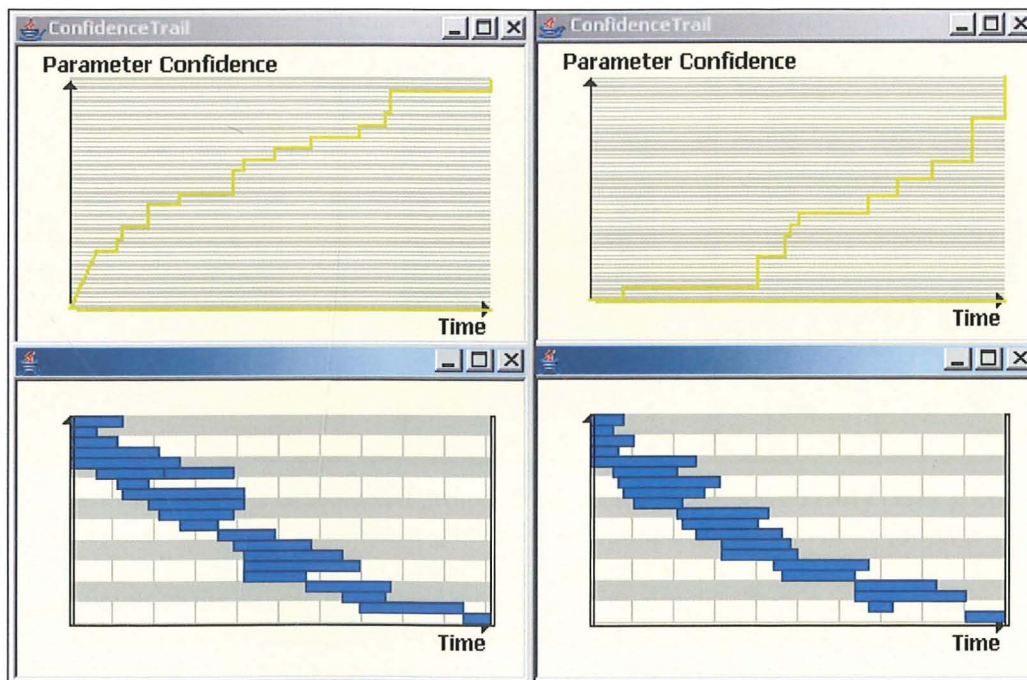


Fig. 7.17 Gantt charts and confidence profiles together provide complimentary information on likely project behaviour. The confidence profile on the top left corresponds to a project in which much confidence is gained early on. The profile on the top right corresponds to a project where little confidence is gained until a significant portion of the project duration has elapsed. However, Gantt chart views on both projects are almost identical, demonstrating the need for both task-based and confidence-based representations together.

Examining the histograms associated with confidence variation and duration variation for the different scenarios considered in section 7.1.4, it becomes clear that variations in process-timing do not necessarily correlate well with confidence-variation (Figs 7.18, 7.19). Thus, failing to account for differences in the way that components accumulate confidence and focusing solely on potential delays can lead to solutions which do not maximise confidence for a given deadline. Product and process should be considered together.

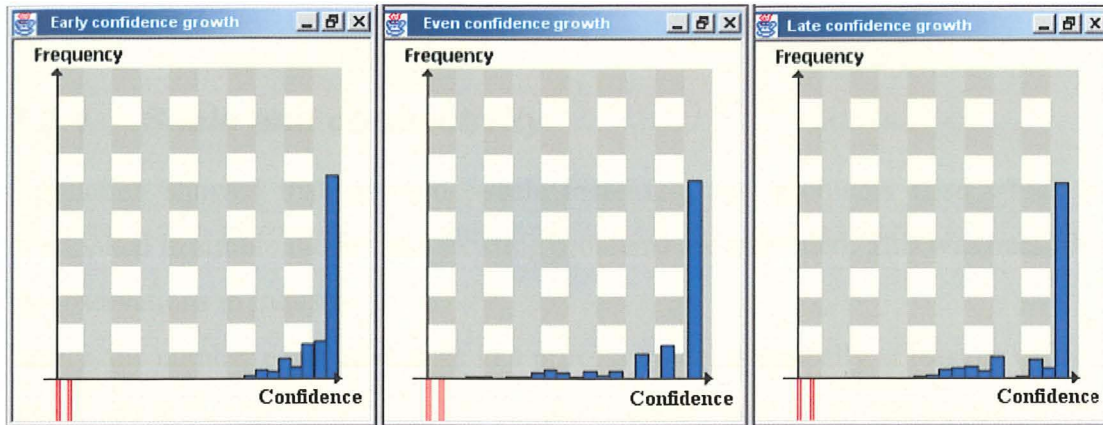


Fig. 7.18 Expected confidence distributions subject to a given deadline (early confidence growth results is more likely to result in high confidence at deadline)

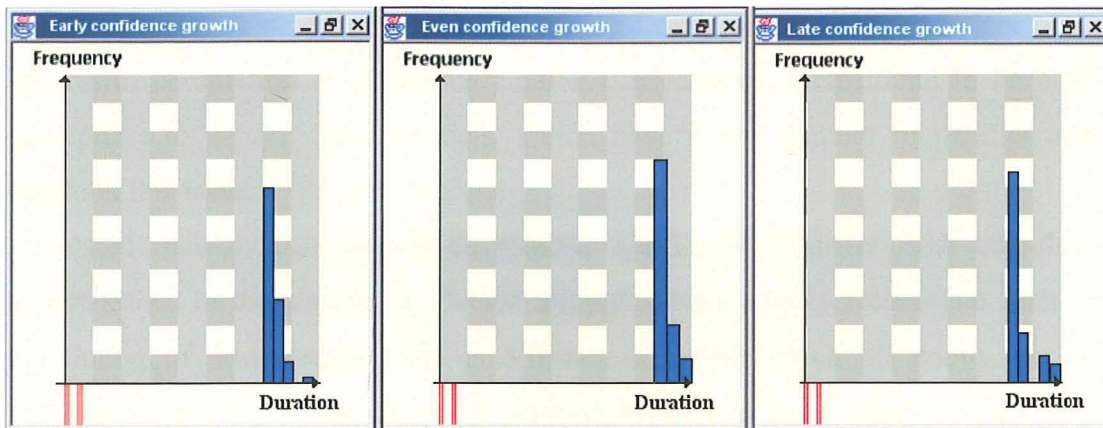


Fig. 7.19 Histograms of expected project duration for projects with early (l.h.s.), even (middle) and late (r.h.s) confidence growth look similar even though these projects vary considerably when examined from a confidence perspective (see Fig 7.18). This observation again points to the danger associated with examining projects from a purely task-based perspective and failing to give appropriate attention to parameter confidence.

7.5 Evaluation of results against real-world data

Evaluation of the above simulation analyses was performed by comparing the results against data from several real-world projects and from interviews with company personnel. During simulation analysis, it is easy to explore the sensitivity to a single

factor by fixing all other project variables at constant values and perturbing the factor of interest. In real-world projects, however, obtaining a clear view of sensitivity to a single variable is more difficult, as controlling other project variables is impossible. In addition, a countless number of models would be needed to completely validate the analysis of the simulation results and such extensive data is not available. Nonetheless, it was possible to investigate whether the trends observed from simulation were reflected in industrial projects. Findings from these investigations are discussed below.

7.5.1 Scale and connectivity

Simulation showed that, as model-scale increases, new problems arise which are overlooked in simple models and existing problems are amplified. This was backed up by observations in industry.

Firstly, the number of parallel tasks in a process model is typically at least an order of magnitude higher in industry than in simple academic models. In large-scale models, where several tasks are on, or close to, the critical path, simulation analysis showed that deviations from the plan are likely. This result was supported by findings from the industrial case study. In particular, testing tasks, which were dependent on the successful completion of numerous parallel predecessors, were found to be highly susceptible to delays. Small models, which have a low number of parallel tasks, overlook this issue.

A related issue is that large-scale projects are likely to suffer from cumulative uncertainties. In the simulation, delays throughout the project accumulate such that the timing of downstream tasks is difficult to predict. Such behaviour was less obvious in industry – the company planned backwards from deadlines as well as forwards based on task durations. Thus the level of delays, predicted by the simulations, was overly pessimistic. However, failure to complete tasks to the desired level of rigour had an impact on product confidence and, from this perspective, errors throughout the project were cumulative. The associated problem of confidence deviations will be discussed in the next chapter (Section 8.3).

7.5.2 Rework

Iteration is acknowledged as a fundamental part of the design process as is evident from the frequent use of feedback arrows in design process models (Chapter 2).

Nonetheless, extensive work describing different rework behaviours and their influence on project performance is sparse. The simulation analysis in this chapter aimed to fill this gap in process understanding.

The results called into question simple models of iteration, which assume that task-properties (such as duration) do not change during rework. They also showed that a fundamental problem with rework is not that it absorbs resources but that it can delay dependent downstream tasks, even when the reworked tasks are performed quickly. Finally, simulation using different task alternatives showed that changing task-order on rework can lead to faster project-completion than performing the reworked tasks in their original sequence.

Discussions with industrial practitioners (and with academics) confirmed that iteration is poorly understood and that simple models of task repetition fail to capture the more complex reality. Sometimes rework is performed in an ad-hoc manner and there is little documentation to show what is actually done, when and by whom – this is especially true of problems which have high urgency and can be quickly solved. In other cases, especially in the event of major failures, the remaining part of the project may be replanned in an effort to meet hard deadlines. This approach corresponded reasonably well with the simulation models that proceeded at higher risk, rather than waiting for new data from the rework of failed tasks.

Insights from the simulation, concerning changing task-order due to rework were also supported by industrial observations; the task-order in rework depends on the cause of failure as does the number of reworked tasks. Signposting's capability to define failures at a parameter level of granularity is fundamental to modelling such process properties; conventional design structure matrices are not capable of modelling at this level of detail.

The simulation analyses also showed that project response to multiple task-failures was context-specific and difficult to predict. This observation was strongly supported by complaints from team-leaders who stated that the real challenge in managing projects is dealing with interactions between task failures. The impact of single failures can be predicted but the way in which multiple failures are likely to interact is extremely difficult to foresee and depends on the problem context.

7.5.3 Confidence

The simulation results showed that the shape of confidence profiles provide clues to the robustness of the process. Components which accumulate a lot of confidence early in the design process are more likely to reach the desired product quality within the project schedule. This was confirmed by observations from industry – components such as the engine block and the head were usually designed early in the process, while the oil-and-cooling system was dependent on information from these components and from customers (e.g. space constraints) and hence matured late in the design process. As a result, failure to meet confidence targets during the oil-and-cooling system design was highly problematic.

The simulation analysis also showed that confidence profiles provide a complementary perspective to solely task-based analyses – even when the tasks associated with the design of different components are similar, the profiles of confidence growth for these components can differ considerably. This observation was confirmed in industry. Perkins was developing a 4-cylinder engine based on an existing 6-cylinder design. Some component designs, particularly those associated with in-cylinder components such as pistons, were allocated a high initial confidence, based on results from the previous engine development programme. In contrast, other components, such as exhaust manifolds, required much design work and were allocated a low initial confidence. Even when both of these components underwent the same engine tests, the confidence contributed to each component varied considerably. Under such circumstances, a task-based perspective provides only a limited impression of the confidence associated with different components and can lead to solutions which are sub-optimal from a confidence perspective.

7.6 Implications of results: heuristics for planning

The above analyses explored process sensitivity to different factors. The implications of these results in terms of defining heuristics for planning are considered below.

7.6.1 Scale and connectivity

During the initial chapters of this thesis, it was made clear that the impact of scale and connectivity on project behaviour is poorly understood in industry and that insights into this issue could lead to improved project planning. Simulation analysis of

hypothetical models, to explore project sensitivity to complexity, led to the below heuristics for planning.

Project-scale should be reflected in the planning approach

Scale matters. Even when the level of task-connectivity is held constant, there are considerably more routes through a large-scale process than through a similar project with fewer tasks – the number of possible task-orders grows exponentially with scale when task-failures are considered. This means that process behaviour prediction becomes exponentially more difficult, an issue which should be considered when planning the project. For example, the use of gateways to reset process uncertainty can be particularly useful in so far as they allow a large-scale project to be decomposed into smaller process-chunks. Alternatively, contingency can be included in plans to allow the process to get back on track in the event of delays of task failures.

A focus on robustness is critical for highly connected processes

Until a saturation point is reached, increasing connectivity increases duration and reduces plan robustness by placing an increased percentage of tasks along the critical path. In this context, it is important that the process is made robust because 1) failed tasks are likely to be critical and 2) rework of several tasks is likely in the event of a single task-failure because the tasks are highly connected. This can be achieved by adding rework discovery tasks, defining non-critical process-recovery tasks or by restructuring the process to remove information dependencies. These issues are especially relevant for concurrent engineering projects which frequently exhibit a high level of task-interdependency.

Overly detailed plans can be inappropriate in the face of high complexity

The simulation results showed that predicting process behaviour becomes considerably more difficult as both scale and connectivity-level increase: small, simple models exhibit low variation and can be planned accurately, while uncertainty accumulates in large-scale models which exhibit high variation in terms of possible routes through the process. As a result, planning using such models is difficult.

The level of detail in plans should reflect the uncertainty associated with process-behaviour predictions. Overly abstract plans will fail to provide sufficiently specific information about process risks, while overly detailed plans are likely to become

unnecessarily demanding from a cognitive perspective. Designers and managers may become frustrated and disillusioned with plans as a result. One approach to dealing with this problem is the use of miracle boxes to represent high-risk process-chunks (see Chapter 4). Further, some parts of the process may be planned abstractly at the project outset and the details defined as new information becomes available.

7.6.2 Rework

The simulation analyses of rework (Section 7.3) led to the following heuristics for planning.

Correct rework modelling is important in creating robust plans

The simulation analyses show how projects can be highly sensitive to rework. In particular, they show that variation in rework probability and different models of task acceleration are likely to have a major impact on the process, and hence that modelling errors are likely to invalidate results. Hence, correct modelling of rework is critical in identifying key process levers and pitfalls. Such knowledge is important when constructing robust plans.

Foreseeable task-reordering, due to late discovery of rework, should be considered during project planning

The simulation results showed that project delays do not result solely from the time spend on reworking tasks but also due to the knock-on changes to the downstream task order (section 7.3.1). Hence it is not sufficient to expedite reworked tasks through approaches such as increasing resource availability; the process must be designed robustly during the initial planning phase. While some process risks cannot be predicted, many can be foreseen. The effects of these failures on the process should be modelled and analysed and, when possible, plans should be created which minimise the resulting process impact.

Plans should account for potential task reordering during rework

In addition to causing changes to downstream tasks, the optimal order of the reworked tasks may differ from the order during their initial execution – depending on the type of failure only a subset of these tasks may need to be reworked, and the design context may have changed in terms of resource availability or learning from previous iterations. The simulation results show that, in some cases, process duration can be

reduced by changing the task-order during rework. In light of this, contingency plans should be developed to recover most efficiently from task-failures.

7.6.3 Confidence

The final heuristics for planning, derived from analyses of confidence data, are presented below.

Confidence should be gained as early as possible

The simulations also showed that confidence should be gained as early as possible in the design process: if failures take place during early tasks, the confidence gap can often be recovered through resource augmentation or by re-planning the process. However, late failures are almost guaranteed to result in expensive rework activities which take place along the critical path and lead to project overruns. One approach to achieving more confidence early in the design process is the use of increased computer modelling instead of physical component testing. In many cases, this has the dual benefit of increasing process robustness while reducing cost. However, in some cases, adequate computer modelling tools do not yet exist and there is no alternative to extensive physical testing.

Product and process should be considered together

The simulation analyses showed that simultaneously examining design progress from both a product and a process viewpoint can lead to insights which can be easily overlooked otherwise. For example, the process that minimises likely delays is not always the same as the process that maximises confidence subject to a tight deadline. Only by concurrently examining the impact of changes in both domains can such trade-offs be better understood. Failure to do so can lead to low product quality or poor process performance.

7.7 Summary

This chapter demonstrated how the simulation analysis of hypothetical design-process models can provide useful heuristics for planning in industry based on structural variations between processes. In doing so it satisfied many of the requirements defined in Chapter 5 (see Table 7.13). It began with a description of the model generator and went on to explore process sensitivity to different factors. It first showed how the behaviour of large-scale, highly-connected models is likely to differ

from smaller, simpler models. It proceeded to demonstrate the importance of correctly modelling rework: failure to do so can lead to poor predictions of project behaviour. Finally, analyses based on variations in confidence were undertaken which highlighted the importance of confidence-growth timing and the benefit of considering the product-process link. The results of the analyses were evaluated based on evidence from the industrial case study and heuristics for planning defined.

While countless other process analyses are possible using the hypothetical model generator and Signposting simulation tool, the utility of the approach has been demonstrated.

Requirement theme	Requirement No.	Achieved	Section
<i>Process sensitivity to scale</i>	<i>S2a</i>	✓	<i>7.2.1</i>
<i>Process sensitivity to connectivity level</i>	<i>S2b</i>	✓	<i>7.2.2</i>
<i>Process sensitivity to task rework duration</i>	<i>R2a</i>	✓	<i>7.3.1</i>
<i>Process sensitivity to task rework probability</i>	<i>R1a</i>	✓	<i>7.3.2</i>
<i>The link between rework and confidence</i>	<i>C4a</i>	✓	<i>7.4.1</i>
<i>Concurrent task-based and confidence based analysis</i>	<i>C3a</i>	✓	<i>7.4.2</i>

Table 7.13 Verification against requirements

8

Lessons from simulation of an industrial project

In addition to structural variations between projects, as discussed in the previous chapter, project-specific variations are also of critical importance during design planning (Chapter 5). This chapter explores how simulation analyses, based on real-world project data, can provide related insights (Fig. 8.1) but also discusses the limitations of the simulation approach particularly in terms of validation.

The chapter first reports on model elicitation in an industrial context and discusses the implications for planning. It proceeds to explore the sensitivity of this model to variation in process rework characteristics before investigating whether (and, if so, how) low-quality completion of tasks early in the design process can cause problems downstream. In considering these issues, the chapter also explores the applicability of the Signposting tool in an industrial setting. The concluding sections focus on the evaluation of simulation results against industrial data and the implications for creating robust plans.

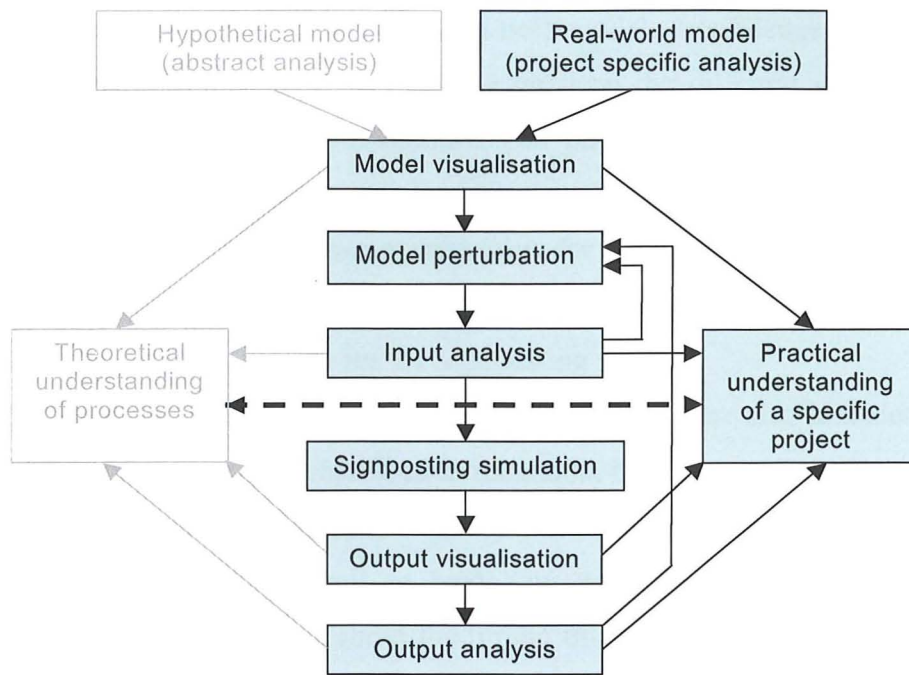


Fig. 8.1 Chapter focus: analysis of real-world models

8.1 Model elicitation

Chapter 5 (Section 5.3.2) highlighted the need to analyse process-specific variations in an industrial model (Requirement S3). Constructing the necessary model was time-consuming and difficult, but nonetheless worthwhile – even in the absence of simulation analysis, it provided useful insights for planning.

8.1.1 Building the Signposting model

Much of the information required to build a suitable Signposting model was already available in existing Perkins documentation. MS Project™ plans, in the form of Gantt charts, showed which tasks would be performed as well as providing information about the task-order. However, they contained no information about confidence or uncertainty. In addition to Gantt charts, Perkins also had an MS Excel™ document which contained rich data on the confidence contribution of different tasks. Nonetheless, the level of granularity in the Gantt charts was different from that in the MS Excel™ charts (Section 4.3) and linking the two data sets to produce the model described in this chapter required roughly 30 hours (Section 8.1.3).

The first stage in constructing the Signposting model involved adding confidence data to an existing Gantt chart. Agreeing on how different Gantt chart tasks contributed to design confidence was a time-consuming and involved undertaking because

irreconcilable task-hierarchies were used in both models (confidence models and MS Project™ plans). Nonetheless, a consensus between the different stakeholders was eventually reached following a combination of interviews with designers and team leaders (see the methodology chapter for interview details), group discussions and off-line conversations, and a common terminology for Gantt chart tasks and confidence tasks was defined. The resulting process map of task connectivity and design confidence constituted the basis for the Signposting model.

Potential future applications of confidence within Perkins were also considered during discussions with team-leaders involved in the model-building process. These included issues such as summing component design confidences to determine system- or engine-level confidence, as well as predicting performance confidence based on project plans. Likewise, issues about the timing of confidence gains were discussed: some tasks only contribute confidence upon completion while other tasks contribute confidence continuously (every hour that an engine test runs contributes to confidence). A flexible modelling framework, which can be adapted to incorporate future developments and applications of confidence modelling, was considered most appropriate in light of these considerations.

After the task-based and confidence-based models had been reconciled, data concerning different process properties, such as rework behaviour (duration of reworked tasks and task-failure probabilities) and task duration uncertainty, was required to complete the model. Part of this information was obtained by examining typical failures and delays on previous projects, as captured by plan-updates, and the remainder from interviews with experienced designers, planners and managers (see Chapter 2). Once all of the necessary data was available, reformatting to create a model suitable for the Signposting tool required a further four hours.

8.1.2 Model description

The resulting Signposting model concerns the development of the oil-and-cooling system for a diesel engine. The model consists of 74 tasks and describes the parallel development of three alternative oil-and-cooling system-configurations – engine mounted, remote and up-rated – for a new engine design (the different configurations reflect divergent customer requirements). The model covers the entire design process, beginning with initial specification for product requirements and concluding with a

validated product. A broad range of activities is covered during the project, including initial design work, performance prediction, Pro-E modelling, drawing, procurement and testing. Many of these tasks are performed in parallel and there is a moderate level of dependency between different tasks. Also, the project was performed to a tight deadline – emissions legislation meant that penalties for late completion were significant. A Gantt-chart description of the model, created using the Signposting tool, is shown below (Fig. 8.2).

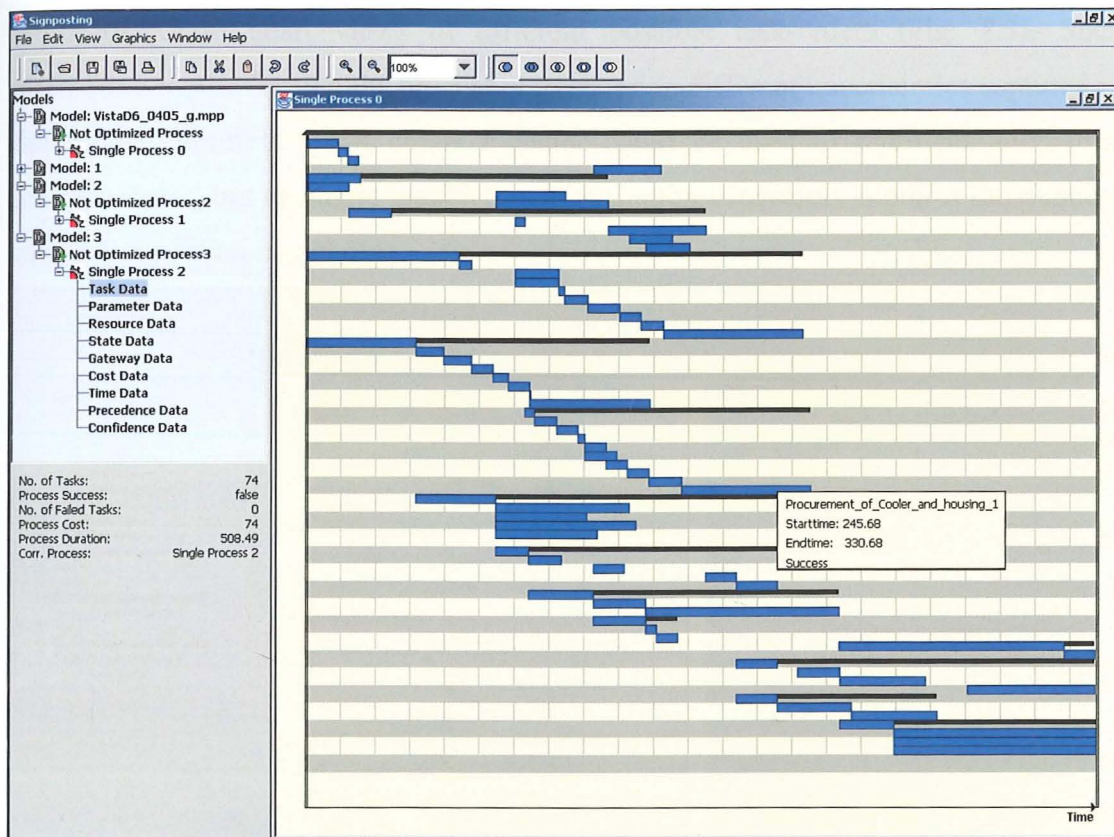


Fig. 8.2 Gantt chart representation of the Perkins model from the Signposting tool

The above Gantt chart representation does not contain information about iteration or alternative process routes. Two outcomes were defined for seven tasks within the process – the first outcome allowed the process to proceed as normal and the second led to rework of dependent tasks. The following tasks drove iteration within the model: Design of Packaging of new cooler (Task 17), Supplier selection (Task 20), Detail drawing of special filter head (Task 30), Detail drawing of new fuel-system components (Task 40), Cold start work (Task 55), DF test (Task 60), 500hr_Gross_Thermal_Cycle (Task 68).

validated product. A broad range of activities is covered during the project, including initial design work, performance prediction, Pro-E modelling, drawing, procurement and testing. Many of these tasks are performed in parallel and there is a moderate level of dependency between different tasks. Also, the project was performed to a tight deadline – emissions legislation meant that penalties for late completion were significant. A Gantt-chart description of the model, created using the Signposting tool, is shown below (Fig. 8.2).

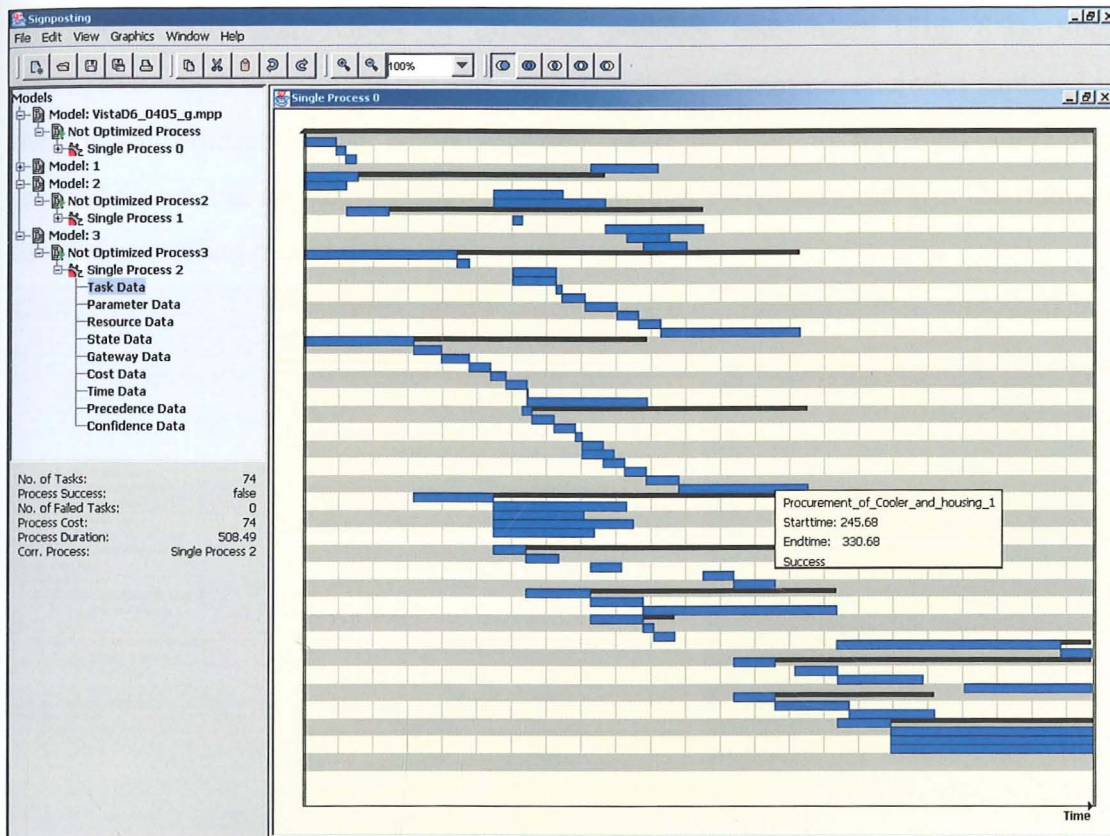


Fig. 8.2 Gantt chart representation of the Perkins model from the Signposting tool

The above Gantt chart representation does not contain information about iteration or alternative process routes. Two outcomes were defined for seven tasks within the process – the first outcome allowed the process to proceed as normal and the second led to rework of dependent tasks. The following tasks drove iteration within the model: Design of Packaging of new cooler (Task 17), Supplier selection (Task 20), Detail drawing of special filter head (Task 30), Detail drawing of new fuel-system components (Task 40), Cold start work (Task 55), DF test (Task 60), 500hr_Gross_Thermal_Cycle (Task 68).

Although seven sources of task-failure may seem low for a 74-task model, it nonetheless constituted a considerable change in the mindset of managers who are familiar with Gantt charts which assume zero-failure. Also, while they acknowledged that other tasks could have failed, they felt that a model that included every possible task-failure would be overly complicated and counterproductive – it was expected that a comparatively simple model was more suitable as a starting point.

The Signposting tool encapsulates design structure matrix views to show iteration alongside Gantt chart views of different possible task-orders (Fig. 8.3). Such alternative routes through the process correspond to different uncertainties captured in the model. Multiple Gantt charts together could be used to represent alternative process routes, but no single graphical representation adequately captures the richness of the Signposting model data.

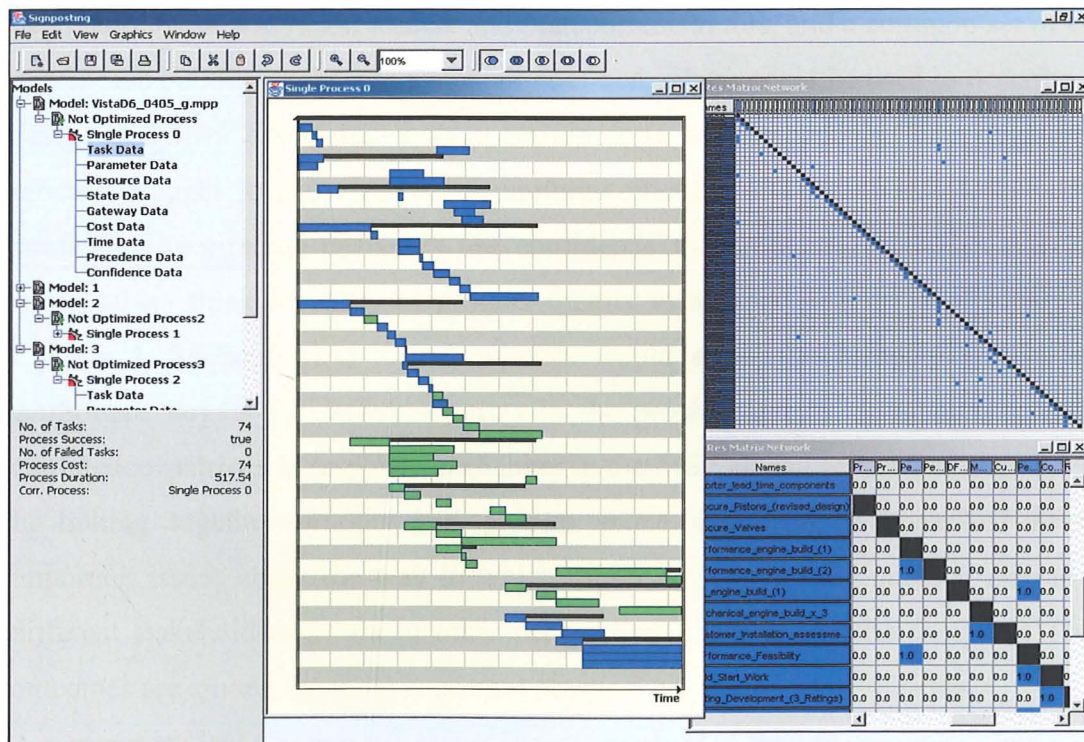


Fig. 8.3 Multiple representation view of the Perkins model from the Signposting tool (shows Gantt chart and DSM views of the process)

8.1.3 Insights from building the process model

Even though the primary goal in model building, from a research viewpoint, was to perform analysis and identify opportunities to create more robust plans, the notion of linking confidence and Gantt-chart plans has direct consequences for Perkins. The manager responsible for project planning (SM) within the company was keen to

integrate confidence data and the more conventional MS Project plans with the goal of creating plans that focused on confidence building activities. As a result, the model elicitation exercise involved some unique challenges but also provided some valuable insights.

This objective of linking the confidence plans and the Gantt charts was initially met with resistance from the design team leaders. Part of the problem concerned ownership: the team leaders had created and continued to administer the confidence plans, while the MS Project plans were administered by the project office (Chapter 3). Further, the confidence matrices were used retrospectively to review progress during project execution rather than to schedule activities, and different terminology was used in confidence plans and Gantt charts.

Differences over the task descriptions were eventually resolved through dialogue and negotiation with both team leaders and plan administrators, and a common set of core confidence building tasks was eventually defined, which could be used to describe the design activities associated with a broad range of components. The negotiation process brought to light hidden assumptions about the different interpretations of confidence (e.g. some designers use confidence as a measure of design maturity, while others think in terms of product quality at a point in time, as discussed in Chapter 4, Section 4.4.4). The consensus would allow the company to establish confidence targets for different points in the process and to automatically update confidence matrices in response to changes in the MSP plans.

In linking together product and process plans this research unearthed several important issues about the way in which diesel engines are designed. It also made different stakeholders aware of other perspectives from within the company. These outcomes are among the most important practical contributions of this thesis.

A summer student (section 2.2.4) was highly involved in linking confidence and Gantt chart plans. In recognition of the value of her work to Perkins, she was awarded third place in the regional Shell Technology and Enterprise Partnership summer-student competition (<http://www.step.org.uk/>). The award was allocated based on the decision of three industrial judges – their appreciation of the work testifies to its practical value. The implications of the model-building process are discussed further in section 8.5.1; the next section discusses insights from the analysis of the simulation input model.

8.1.4 *Insights from analysis of the simulation input model*

Even before any simulation analyses were carried out, insights into the likely process behaviour were obtained by examining the Signposting-simulation input-model. The model contains information about the number of high-risk tasks, the level of connectivity between tasks, the level of uncertainty associated with task-duration and also the estimated cost. By considering these factors in combination, predictions about the process behaviour were made.

For the model described above (Fig. 8.3), the average level of connectivity between tasks (filling grade for a DSM) is low compared to published work on DSMs, but there are a few tasks which have many dependencies. Also the degree of parallelism is reasonably high (roughly six tasks in parallel). Because of this high level of parallelism, it was inferred that if tasks fail, the impact of rework will be significant because it is likely to be on the critical path. Further, there are some tasks (e.g. testing tasks) which are dependent on several predecessors. Hence, delays to these tasks are likely because of the high number of process execution scenarios which can impede their progress. As these tasks lie on the critical path, such delays will affect the process performance.

While looking at the process input model gives some indication of its likely behaviour, it yields little insight into process sensitivity to different uncertainties. In contrast, the combination of model perturbation and simulation provides a powerful means to explore such issues, as described in the following sections.

8.2 Variations due to rework

Process plans are forward-looking, and hence are based on estimated values. In order to build models, which constitute a reasonably accurate prediction of the way in which the process will unfold, it is useful to understand process sensitivity to different sources of error. Otherwise, effort may be wasted in gathering very accurate estimates for variables against which the process is reasonably robust, while rough estimates may be unwittingly used for more critical variables. Knowledge of sensitivity to different variables can also be used to improve the process. This section explores process sensitivity to rework probability and to task duration during rework.

8.2.1 Rework probabilities

Requirement R1b (Section 5.3.3) identified the need to explore process sensitivity to rework probabilities. The previous chapter examined this issue based on hypothetical models. Further analyses based on a real-world model are discussed below.

Estimating the probability that a task will fail and require rework is a difficult problem as reported in the literature (Yassine et al., 2001) – even when similar tasks have been performed previously, the changing task-execution context is likely to change the probability that these tasks will fail again. Simulation analyses were undertaken to determine process sensitivity to rework probabilities.

To this end, the model described in section 8.1.1 was perturbed such that task-failure became firstly 100% and then 200% more likely (for seven high-risk tasks) and 1000 simulations were run. While a 200% increase in the likelihood of task-failure may seem excessive, this change corresponded to changing the task-failure probability from 10% to 30%, a change which was considered possible for the most risky tasks in the model. Even though 30% failure likelihood was thought to be quite pessimistic, they were still interested in seeing the implications for the project. The results for the mean and standard deviation for project duration are shown in Table 8.1 and Fig. 8.4.

	Mean (days)	Standard Deviation (days)
No Rework	615.5	0
Low rework probability	634.9	20.4
Medium rework probability	641.6	24.9
High rework probability	648.2	32.3

Table 8.1 Project sensitivity to different rework likelihoods

In addition to confirming that both the mean and the standard deviation for the project are positively correlated with increasing levels of task-failure, the analyses also show that the presence of rework makes predicting the likely task order during project execution more difficult. For low levels of rework, the project duration corresponding to the bin on the top, left-hand-side of the histogram in Fig. 8.4 predominates. Project planners can be confident that the task order(s) associated with this duration is appropriate. As the level of rework increases, however, the higher standard deviation indicates that the process is unlikely to have a single outcome in terms of duration. Several different process execution durations become almost equally likely,

particularly in the case of 30% task-failure likelihood (Fig. 8.4). This makes planning difficult because of the wide number of process routes which must be considered.

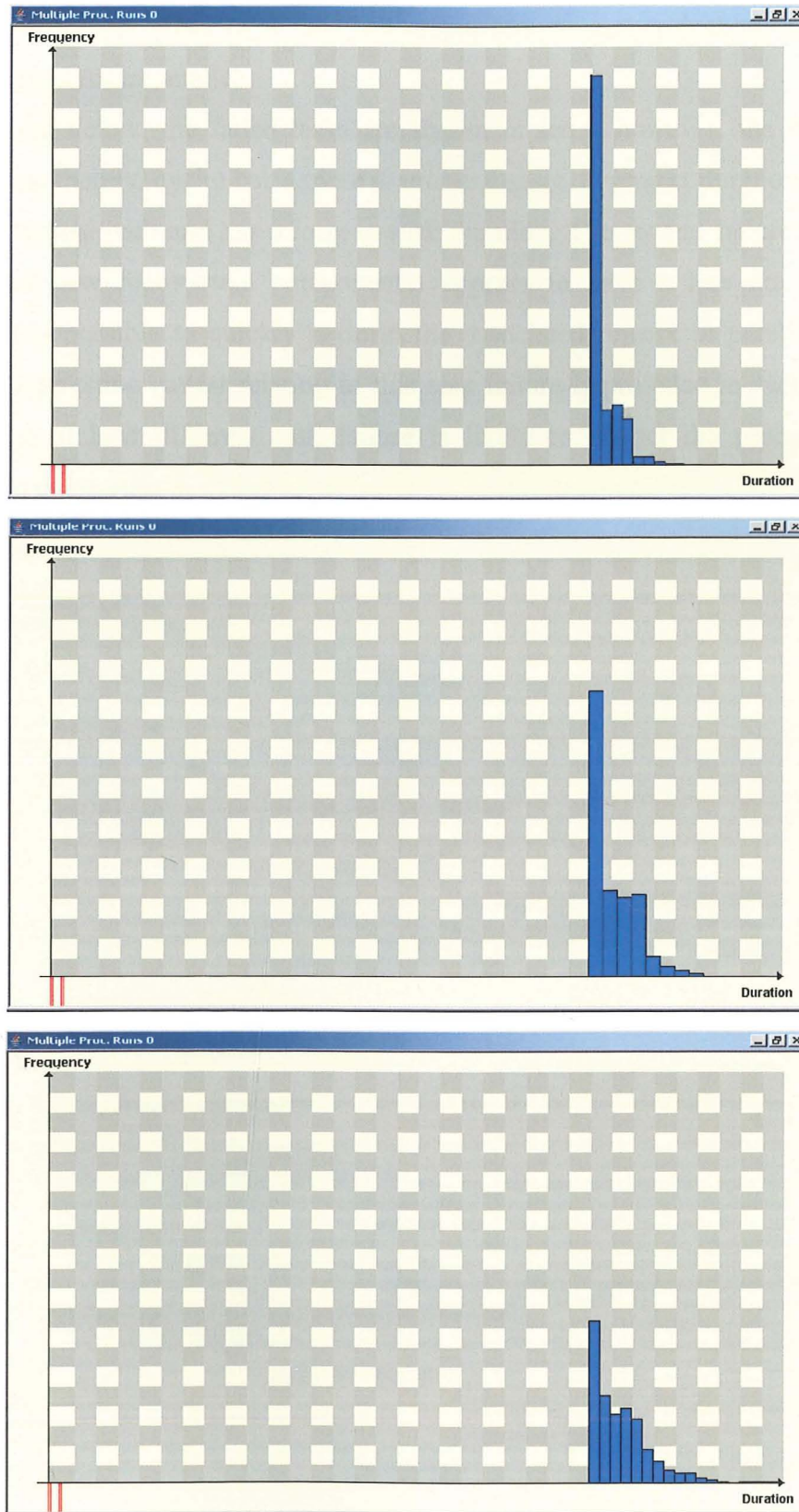


Fig. 8.4 Histograms for different task failure likelihoods (10%, 20% and 30% from top to bottom) show how project duration rises due to rework

Looking more closely at the individual simulation runs, it can be seen that different task-failures affect the process differently (Fig. 8.5) – some failed tasks are on the critical path, others close to it, and yet others remain non-critical unless multiple failures take place in parallel.

In the figure below, the failed tasks are shown in red – however, the first failure (500hr gross thermal cycle) has a more dramatic impact on project duration because it is on the critical path and has a long rework duration. The second failure (the detail drawing of new fuel-system components) appears to have a less drastic impact because it is possible to quickly perform the associated rework in parallel to other tasks (an interesting caveat relating to this task failure is provided in section 8.3.2). Knowledge of how different task failure is likely to impact the process can be valuable to companies in defining strategies for risk-mitigation.

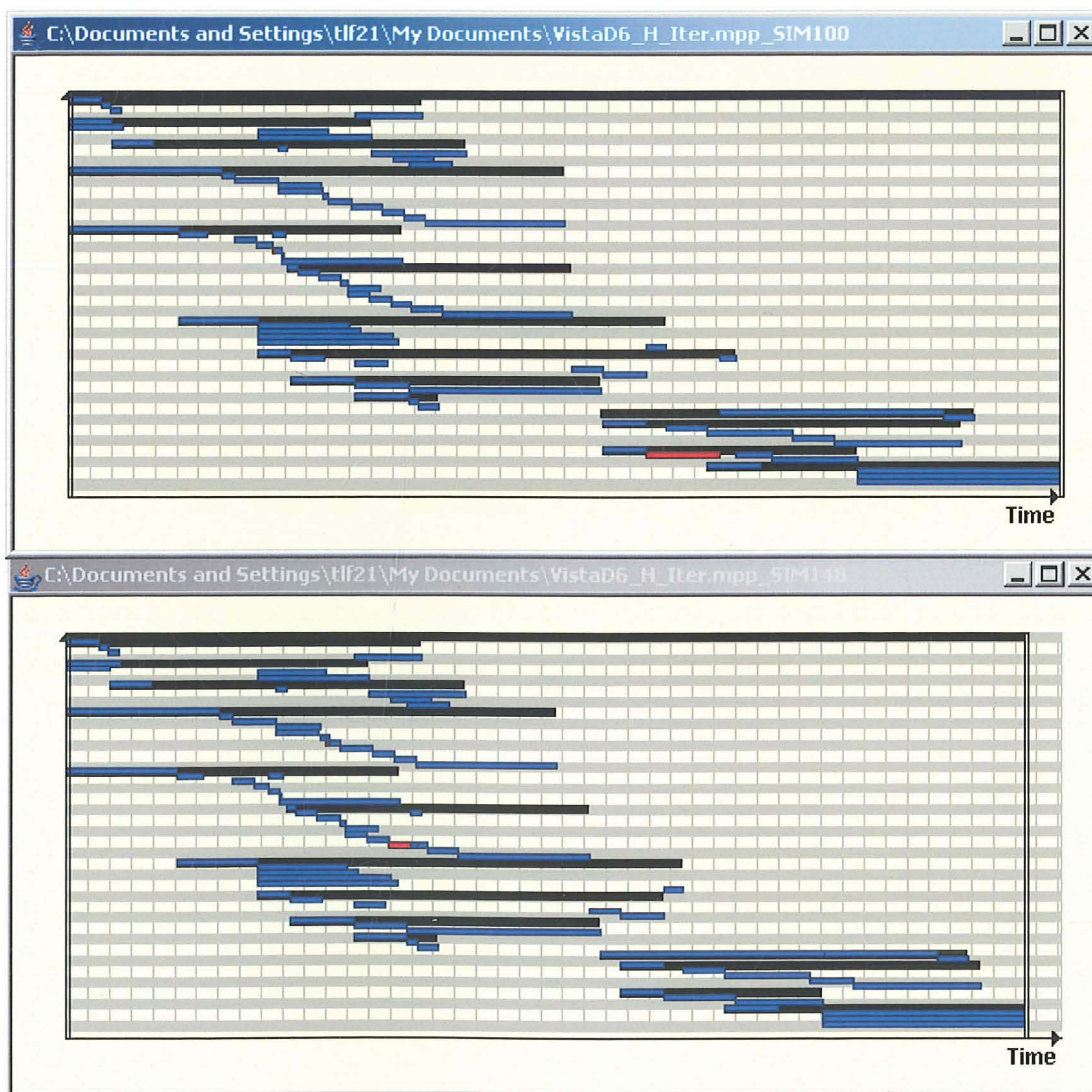


Fig. 8.5 Different task failures shown in red have a different impact on project duration

8.2.2 Rework duration variation

In order to investigate the impact of rework duration variations in a real-world model (Requirement R2b, section 5.3.3) further simulation analyses were performed. To this end the model described in section 8.1.1 was again perturbed by respectively increasing and decreasing the estimated task durations during rework by 50% and 1000 simulations were run. The results for the mean and standard deviation for the project duration are shown in Table 8.2 and a scatter plot of the different simulation results is shown in Fig. 8.6.

These results corroborate the findings in section 7.3.1 concerning the hypothetical models. However, the high number of task precedences in the real-world model meant that less reshuffling of downstream tasks took place.

	Mean (days)	Standard Deviation (days)
Slow rework	625.3	22.6
Rework at anticipated speed	622.5	14.4
Accelerated rework	620.3	9.8
No rework	615.5	0

Table 8.2 Project sensitivity to task duration during rework

The impact on project performance, in response to perturbations of task rework-duration, is less than that associated with the earlier changes to the rework probabilities. In particular, the sensitivity of the duration-mean is low. This implies that errors in estimating the duration of reworked tasks are less likely to lead to poor estimates for project duration than errors in rework likelihood.

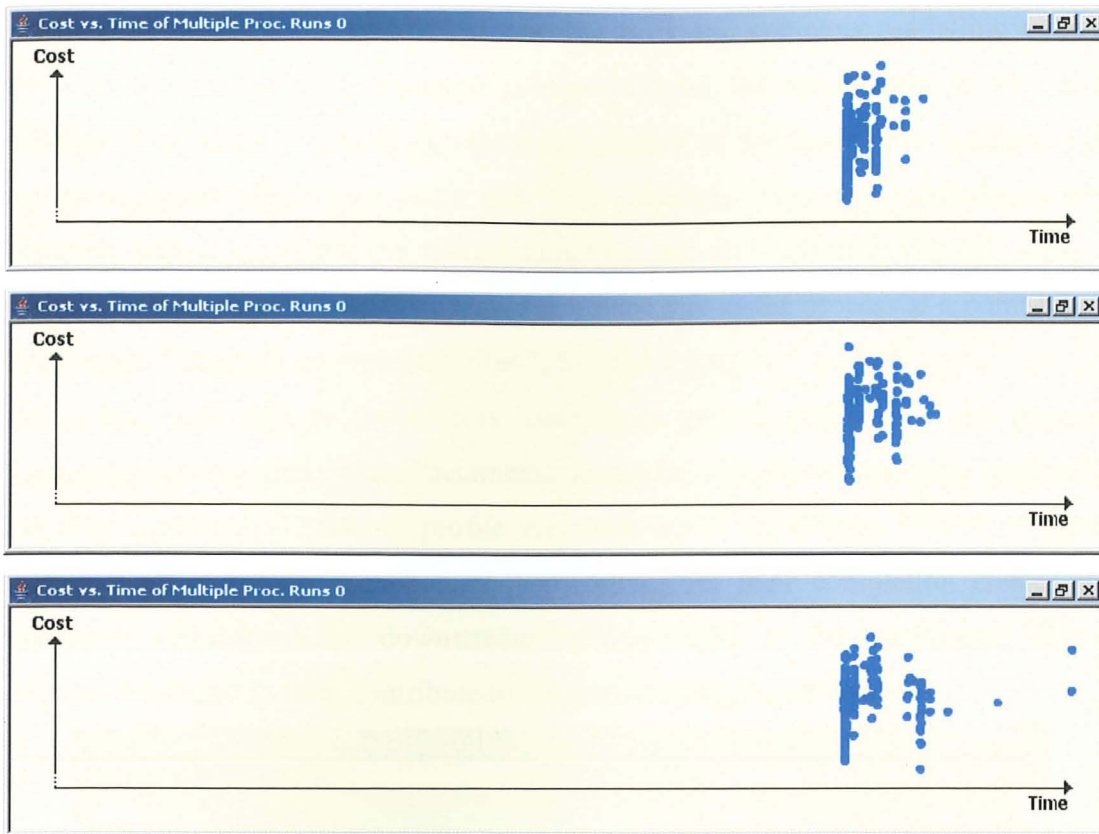


Fig. 8.6 Scatter plots for different task-rework durations, drawn to the same scale show that increasing the duration of reworked tasks increases the mean and variance for project duration

8.3 Variations due to confidence: a product-process perspective

Product-driven decisions can have a major impact on process tasks. However, the link between the product and its design process receives little attention in literature on planning even though resulting oversights can lead to major project delays. The use of confidence profiles, to explore the product-process link based on actual project data, is discussed below.

8.3.1 Confidence and task-prioritisation

This section addresses requirement C3b (Section 5.4.4) – the need to concurrently examine processes from a task-based and a confidence perspective. Confidence profiles show which tasks contribute the most confidence and also show when, during the process, confidence is obtained. For the oil-and-cooling system, three different designs are being developed - engine mounted, remote and up-rated. The confidence associated with these three design configurations was tracked throughout the

processes. The model shows that early design work and computer modelling tasks at the end of the process contribute substantially to the confidence in the three configurations (Fig. 8.7). Also, a significant portion of the confidence is contributed by testing tasks which take place late in the process. Such tests included a heat rejection test, a noise test, an oil consumption test and different thermal tests. In contrast to the design work, these tasks take place late in the process at a point where failures are difficult to recover (Chapter 7, Section 7.4.2).

About half way through the process, confidence growth stagnates – this happens during the procurement of test specimens, as can be seen by concurrently examining the Gantt chart and confidence profile visualisations of the process (Fig. 8.7). These tasks do not explicitly contribute to confidence, but their completion creates the necessary conditions for downstream testing tasks to be performed. These downstream tasks, in turn, contribute to the overall design confidence.

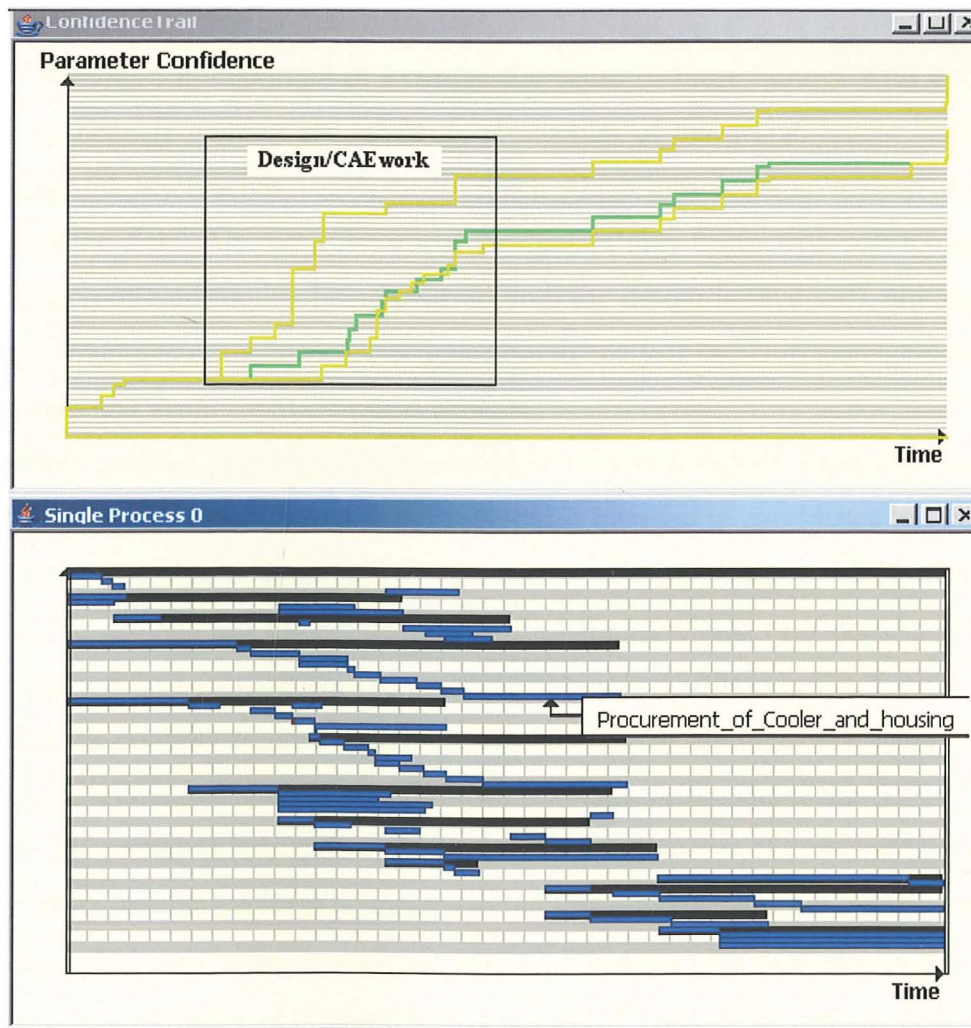


Fig. 8.7 Process analysis using confidence profiles and Gantt charts (much confidence is acquired during design/CAE work)

The slope of the confidence profiles represents the confidence contributed per unit time – examining the slope of the confidence profiles in Fig. 8.7, it can be seen that design and computer analyses, performed at the start of the process, contribute much confidence over a comparatively short time period. This suggests that increased computer-based testing, as an alternative to more costly confidence-building activities such as physical testing, can lead to process improvements.

8.3.2 Confidence and rework

The final requirement of Chapter 5 – the need to explore the link between confidence and rework at a process-specific level (Requirement C4b, section 5.4.4) – is met by the analyses in this section. Section 8.2.1 discussed rework due to a task failure in the 500hr gross thermal cycle test. Due to this failure, and the corresponding rework, the process duration is 12% (just over two months) longer than the zero-failure simulation. In the Perkins design context, such a delay to engine delivery is not an option. Hence, it is important to examine the cause of these failures with a view to avoiding them. If nothing can be done, production of the design has to commence before confidence reaches 100%, an approach which introduces an undesired risk of failure in the field. Alternatively, initial production may begin with soft tooling (the introduction of hard tooling being delayed until the design is finalised), although this increases manufacturing costs. Confidence-based analysis can help managers improve the process and avoid such additional expense.

Confidence profiles also highlight the problem of partial failure. Partial failure takes place when a task fails to provide the anticipated increase in confidence but does not result in a confidence drop, and is difficult to identify from a solely task-based perspective. Comparing the confidence profiles for simulated runs corresponding to success and failure (respectively represented in blue and yellow in Fig. 8.8), it can be seen that the confidence profiles diverge due to a partial failure of the detail drawing of new fuel-system components task (shown in orange on the Gantt chart).

At first, the partial failure appears not to be severe. However, the second task-failure (500hr gross thermal cycle) is connected to the first – the reduced confidence resulting from the initial task-failure causes the execution context of a downstream dependent task to change in terms of confidence and ultimately leads to the second task-failure, which results in rework and process delays.

When detail drawing of new fuel-system components fails to deliver the expected confidence and the resulting rework is performed quickly, the project can precede relatively undisturbed (Section 8.2.1). Conversely, failure to quickly address these problems can lead to major rework. The capability to identify the drawing activity as a process lever puts Perkins in a better position to avoid the 2-month process-delay.

Partial failures are especially problematic when they occur in combination – if several tasks, which affect the same parameter result in small confidence shortfalls, the cumulative confidence deficit is likely to be problematic, but each task may be considered successful in its own right. Divergence in confidence profiles, however, immediately flag up confidence gaps due to the partial failure. Also, the size of this confidence gap indicates the severity of the failure, thus helping planners and managers prioritise different risks (Fig. 8.8).

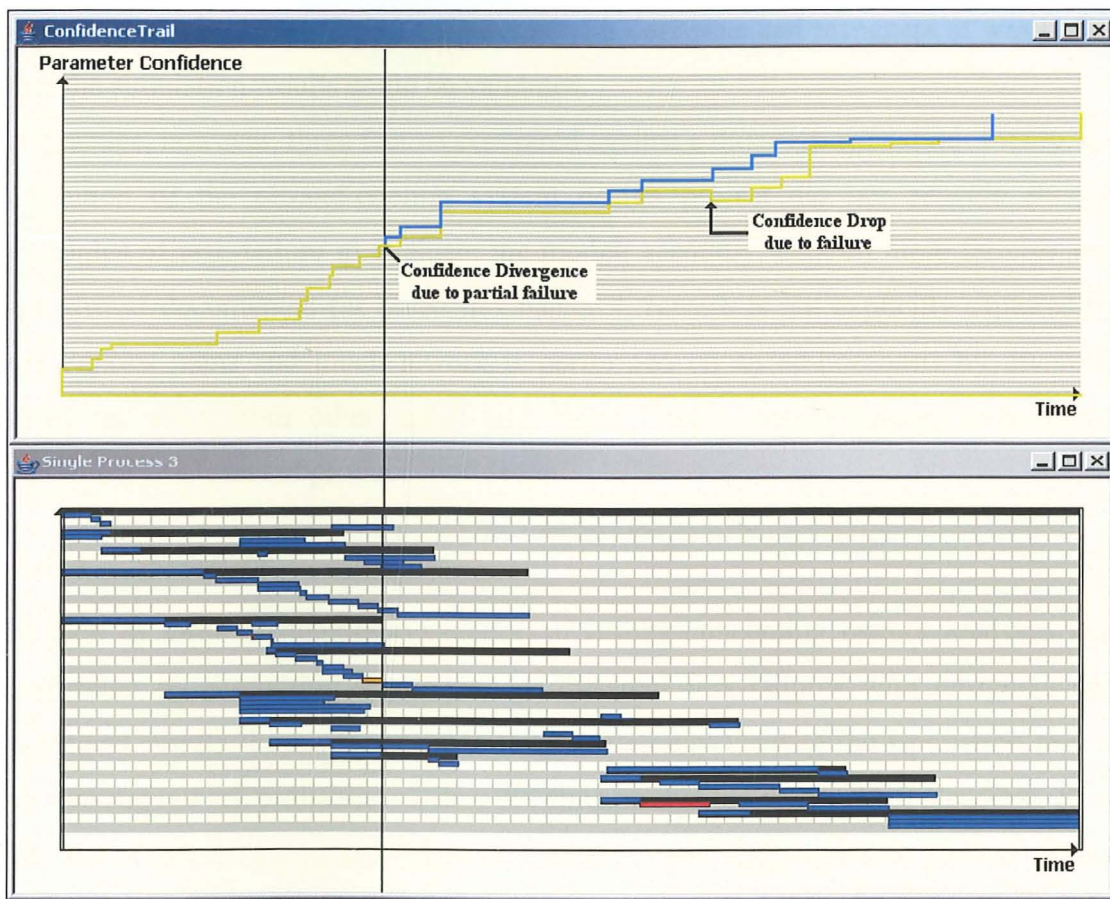


Fig. 8.8 Divergence in confidence profiles due to partial failure (the confidence profiles highlight the potential downstream rework while the Gantt chart remains unchanged)

8.4 Evaluation of simulation results

This section compares observations during project execution in industry to those predicted by simulation analysis. In doing so, it highlights the merits and limitations of the Signposting approach to model-simulation.

The simulations described in this chapter, suffer from a common limitation of all socio-technical systems: they are inherently very difficult to validate (Johnson, 2001). Even if the same project were performed repeatedly, it would not be possible to completely validate results due to learning during different project runs. It is, however, possible to partially validate results by looking at historical data on task-failures and delays in Perkins and examining the resulting impact on the process. Perkins performs plan updates on a weekly basis. During these updates, information about task-failures and project delays is added to the plan. By examining such updates, and discussing their implications with designers and managers, it was hoped that the simulation results could be validated.

In practice, however, three factors conspired against a thorough validation: Firstly, the execution of the oil-and-cooling system design project did not provide enough examples of task-failure to evaluate the model; secondly, data on these failures was not always documented (as noted in Chapter 4, rework is not explicitly captured by the Perkins plans and much rework is performed on an ad-hoc basis by diverting resources away from other tasks) and, thirdly, data on task-failures is sensitive, particularly in terms of product confidence.

8.4.1 *Rework*

In light of these difficulties, validation of results concerning rework analyses was performed by discussing the impact of different task-failures with designers and managers. While they were not able to quantitatively validate the simulation results concerning rework probabilities, they did confirm that the results were plausible. They also noted that the predicted project delays looked slightly pessimistic (tight deadlines demanded on-time completion) but the anticipated variations would instead be reflected in terms of design confidence. They confirmed that increasing levels of rework detracted from the plannability of projects, citing examples of rework due to changes in other components, or external requirements, which were extremely difficult to plan for. For example, changing engine-block requirements in response to

different customer desires complicate the planning and management of the oil-and-cooling system-design (due to component dependencies).

When examining the simulation results, it became clear that the impact of different failures on the process varied significantly. While structural-level issues, such as the timing of high-risk tasks, provide insights into likely process behaviour (see Chapter 7), process variations are also possible due to the specific nature of task failures. The simulation results in this chapter show how factors, such as confidence shortfalls, degree of rework required and the interaction with other task failures, can all play an important role. Perkins agreed that the impact of different failures was varied and difficult to predict and that planning for such failures was highly challenging. For examples, procurement delays were likely but had a low rework impact, while test failures were less likely but generally had a severe rework impact.

Validation of the specific results concerning rework duration uncertainty was also problematic due to the low number of failed tasks experienced in the real-world project. In particular, there were few incidences of slow rework – most reworked tasks were performed quickly so that downstream tasks could proceed. Also, there was a tendency within the company to stick to the plan, where possible, a factor that is not captured in the simulation. In addition, it was noted from discussions with managers that some rework was disguised as delays to other tasks.

Nonetheless, the overall results from the simulation analysis were supported by comments from interviewees. The first result was that project sensitivity to rework probability was likely to be higher than sensitivity to task-rework duration. Managers contended that it is more difficult to predict when rework will happen than it is to predict how long it will take if it does happen because the rework-duration can often be influenced, for example through resource augmentation. The second result was that changes to the task-order mattered more than the time spent reworking failed tasks. In practice, Perkins designers do not stand by and wait while other tasks are being reworked. Instead they proceed with other parts of the process subject to a lower level of confidence. This approach can result in major deviations from plans.

During project execution, some events took place, which were not predicted by the model. For example, delays to one engine test occurred due to late procurement of other components. Such limitations of the model were expected, as complete knowledge of all possible problems cannot be known in advance. Nonetheless, the

applicability of the Signposting simulation approach, to real-world project analysis, was demonstrated.

8.4.2 Confidence

The simulation results show how confidence profiles can be used to prioritise different tasks in terms of their confidence contribution. They show why process efficiency can be improved by shifting the design work away from physical testing and, instead, increasing the amount of computer-based validation. This insight was confirmed by observations in industry – due to recent advances in specialised modelling software, the electronic design and validation of components such as crankshafts is becoming increasingly popular, reducing the necessity for physical testing.

Results relating to partial failure and rework were also supported. Although it is clear that delays to activities on the critical path will lead to project overruns, determining how non-critical tasks can delay the process is challenging. As discussed in section 8.3, however, small variations in the confidence contribution of non-critical tasks can lead to delays in dependent, downstream tasks due to product-process interdependencies or rework. This issue was also observed in industry when a decision was made to procure engine-blocks from a new supplier. Although no changes to the block-design were desired, the manufacturing processes used by the new and old suppliers differed, resulting in dissimilar residual stress patterns within the block. Later test-failures were eventually traced back to this issue.

While the confidence-based results were supported by industrial findings, some discrepancies were noted between the confidence profiles from the simulation analyses and those created as the project unfolded. This happened because the simulation model only accounted for confidence growth upon task completion, while designers frequently claimed increased confidence as the tasks progressed (see section 4.4.4). Although such data could be added to the model, it is unlikely that the potential benefits would justify the increased model complexity.

Overall, the utility of the tool, as a means of providing insight into likely scenarios and guidance on suitable strategies to reduce their impact, was shown. The confidence-based analysis discussed in this thesis sparked useful discussions about the types of failure which eventually took place. Such discussions were beneficial to managers and designs alike by forcing them to think through the different possible problem-causes and their consequences.

8.4.3 Limitations

Despite its utility, simulation using the Signposting tool is not without limitations. Firstly, the analyses are limited by the quality of the project model; errors and omissions in the latter can lead to inaccurate results. Further, agreement with a single industrial project in a single industrial sector does not constitute complete validation of a tool; major disparity between simulated data and actual events would, however, raise concerns about the soundness of the approach. In this light, the results reported here should be viewed as encouraging, but further work is needed to extensively validate the tool. Finally, building the models and interpreting the simulation results is non-trivial: while significant extensions have been made to the tool to simplify this process, further enhancement to the tool is a worthy candidate for further research. (Although, Wynn et al. (2006) have developed a useful tool to support model-building, his treatment of confidence is considerably different from that reported in this thesis and substantial changes to his tool would be required to create the kind of models analysed here.) Nevertheless, the results reported above, coupled with feedback from designers and managers in industry, testify to the tool's utility.

8.5 Implications of results: heuristics for planning

The above results have implications for planning in terms of 1) modelling and representation, 2) rework and 3) the product-process link. Opportunities exist to increase process robustness, and hence improve the quality of plans by focusing on any combination of these themes.

8.5.1 Modelling and representation

This section discusses the implications of this research for modelling and representation. These issues are critical because they act as the foundation for all subsequent analysis.

Model building efforts should be aligned to industry's needs

Perkins was interested in linking plans to confidence data, not just for the purposes of this research but as a company-wide strategy for improving the quality of their plans. As a result, personnel within the company thought intensely about the practical issues of modelling, bringing hidden themes and assumptions to the surface which would be overlooked by a "quick and dirty" model-building exercise. Even in the absence of

any simulation analysis, the resulting negotiations highlighted opportunities to create better plans, by catering for the needs of different stakeholders. Some researchers present new analyses based on data from previous case studies (for example, Cho and Eppinger, 2005). This research showed that, while performing extensive case studies is more challenging, the rewards, both for the company and the researcher, are considerable.

Modelling focus should reflect process sensitivities

The simulation results demonstrate that process sensitivity to different uncertainties varies considerably. For the above models, errors in rework probability estimates are likely to invalidate results, while small errors in task-rework duration-estimates are tolerable. Thus, the analyses show which errors are likely to predominate (errors in rework-probability estimates) and hence where to focus efforts during modelling. Appropriate focus can improve model-quality while reducing the time and effort required for model elicitation. High quality models lead to improved process analyses, increased process robustness and better plans.

Design process models should account for rework

Both the model building and the analyses phases showed that the content of the models affects the way in which people think about processes. People analyse models as opposed to processes and are hence constrained by the limitations of their model. While several types of plan are commonly used in Perkins, none of them show iteration or rework. As a result, problems associated with rework are non-transparent. This work showed that rework is an important issue in Perkins. While designers and managers were initially hesitant to acknowledge rework, discussions about how confidence grows during design execution brought hidden assumptions about rework to the surface. For example, designers assumed that changes to the product would be required as the manufacturing process was finalised and that the resulting rework would be performed by diverting resources away from other tasks. However, these issues were not captured in Perkins' process plans, a factor which sometimes led to the adaptation of a fire-fighting approach.

Multiple representations should be used together

The model building and analyses phases of this research also demonstrated the benefits of using multiple visualisations. The Signposting tool for process analysis allows the designer to concurrently examine multiple views of the same project. Gantt

charts, design structure matrices (DSMs), domain mapping matrices (DMMs), scatter plots, networks and confidence profiles can all be used in combination to interrogate processes. Each representation shows different information: for example, Gantt charts contain rich data on task timing, while DSMs show how different tasks are connected together. The amount of information associated with a complex design project is too large to be displayed in one single visualisation: multiple, interlinked representations, however, realise the visualisation of complex information in manageable chunks.

8.5.2 Rework

Failure to adequately model and analyse rework reduces process robustness and detracts from the value of plans. Steps should be taken to avoid rework and, when this is not possible, to reduce its impact.

Process analysis for rework susceptibility is beneficial

Iteration is complicated and poorly understood: many models ignore iteration and rework completely; others such as DSMs acknowledge the existence of iteration but provide little insight into the manner in which it is likely to affect processes.

The simulation analyses, presented earlier in this chapter, explored how rework (iteration due to task-failure) was likely to impact process duration. Results showed that process performance is highly sensitive to both task-rework duration and rework probability. The analysis also identified the tasks that are likely to drive major rework cycles. Knowledge of these tasks is valuable in defining strategies to avoid rework completely or to reduce its impact on the process.

Steps should be taken to avoid rework

Once the tasks that potentially drive rework have been identified, these tasks can be modified to reduce their failure probability. For example, more time or resources can be made available to increase the quality of information produced by such tasks. This corresponds closely to the approach taken within the company – high-risk parts of the design are dealt with by six-sigma focus groups of experts.

Another approach is to increase the margin associated with some component designs so that they can absorb changes that would otherwise lead to rework. Alternatively, more modular designs, which exhibit higher degrees of functional independence, (Suh, 2001), can help designers avoid rework because changes to different parts of the design are less likely to propagate.

Processes should be made more robust against rework

If rework cannot be avoided completely, steps can nonetheless be taken to reduce its impact on the process. For example, if rework can be planned such that it takes place in parallel to the critical path, then its impact on the process decreases significantly. Further, as described in Chapter 5 (Section 5.3.1), appropriate task-ordering can reduce the amount of rework resulting from unchanged levels of task failure.

In some cases, the introduction of new tasks, aimed specifically at finding rework, can reduce the time taken to discover rework and hence improve process performance (Cooper, 1993). Similarly, the use of confidence profiles to track gaps between the predicted and actual confidence values can help designers and managers detect rework early on. Once rework has been discovered, appropriate actions can be implemented to make the processes more robust.

8.5.3 The product-process link through confidence

The product-process link lies at the core of design. Nonetheless, much analysis is domain-specific and this link is under-researched. The implications of the product-process link for designing robust processes and better plans are discussed below.

Confidence data should be used to prioritise tasks

Conventional process plans, such as Gantt charts, fail to show how tasks vary in terms of the confidence that they contribute to the design. During the design of the oil-and-cooling system, some tasks (e.g. testing) contribute considerably to the overall design confidence while others have a comparatively small effect. Even with tasks, which look the same in terms of duration, cost and resource utilisation can vary greatly from a confidence-perspective. When planning projects with tight deadlines, prioritising tasks based on their likely confidence contribution can maximise the design confidence expected subject to given time, cost or resource constraints.

Plans should account for potential confidence shortfalls

The final implication from the analyses is that plans should account for undesired task-outcomes which result in confidence shortfalls. Confidence profiles show that small confidence shortfalls (partial failures) can adversely affect the execution context of dependent tasks (these tasks are executed based on false assumptions about confidence), causing confidence deviation to escalate. Whether the process is derailed by the confidence shortfall or whether it recovers varies on a task-specific basis.

Even when these failures do not explicitly drive rework, they can nonetheless result in major reshuffling of downstream tasks which is harmful to the process because it causes delays and deviations from the plan. If, on the other hand, they do act as rework drivers, the implications for the processes are even more severe. Confidence profiles can help designers and managers identify the tasks and task-combinations which have a major impact on the process. Plans can then be constructed which are designed with the goal of prompt recovery from confidence shortfalls (through the use of contingency, resource augmentation, appropriate task-ordering). Such plans are more robust in the face of uncertainty.

8.6 Chapter summary

This chapter evaluated the Signposting simulation tool using data from an industrial project with a focus on 1) modelling and representation, 2) rework and 3) the product-process link. It described the model elicitation process and the simulation analyses performed. Even in the absence of the subsequent simulation analyses, the modelling exercise led to an improved understanding of process behaviour. The simulation analyses highlighted the importance of correctly modelling rework and demonstrated how confidence data can be used to identify rework quickly hence minimise its impact. Simulation results were compared against actual events during project execution in industry and reasonable agreement was observed. The implications of the analyses in terms of creating better plans were presented. The chapter also satisfied the outstanding requirements defined in Chapter 5 (Table 8.3).

Requirement theme	Requirement	Achieved	Section
<i>Process-specific model variations</i>	<i>S3</i>	✓	<i>8.1</i>
<i>Process sensitivity to task rework probability</i>	<i>R1b</i>	✓	<i>8.2.1</i>
<i>Process sensitivity to task rework duration</i>	<i>R2b</i>	✓	<i>8.2.2</i>
<i>The link between rework and confidence</i>	<i>C3b</i>	✓	<i>8.3.2</i>
<i>Concurrent task-based and confidence-based analysis</i>	<i>C4b</i>	✓	<i>8.3.1</i>

Table 8.3 Verification against requirements

9

Conclusions

Many PhD projects involve an intellectual journey of discovery where the researcher sets out without a specific destination in mind. Even when the destination is known, the route is often unclear. As time passes, different events unfold and the researcher is signposted towards the next interesting piece of research. Eventually, as time and/or funding run out, a subset of the interesting discoveries is selected and these ideas are moulded into a coherent story, which omits blind alleyways that have been explored, pieces of research that do not fit with the thesis storyline, and ideas that remain underdeveloped. If the research path can be thought of as a spiral, where the research moves slowly towards the centre, revisiting different themes to reflect a changing information context, the thesis can be pictured as the straight line from the outside to the inside of the spiral.

This chapter first reflects on the findings reported throughout this thesis; it outlines the core conclusions, highlights research contributions, evaluates the work against the research questions defined in the introductory chapter, and notes limitations. In addition, opportunities for further work – those underdeveloped ideas – are discussed.

9.1 Core conclusions

- Design planning challenges arise when representing uncertainty and connectivity in large-scale plans and when linking information from several different process models. Modelling product-process interdependencies and planning for rework are especially difficult. (Chapter 4).
- Currently, the tacit overview of experienced designers plays an important role in overcoming the limitations of plans. Relying on overview is likely to prove problematic as experienced personnel retire or move on against a backdrop of rising product and process complexity (Chapter 4).
- The analysis of process structure using hypothetical models provides insights into process behaviour but avoids fixation and bias problems which arise when examining real-world models (Chapters 6 and 7).
- The plannability of large-scale, highly-connected models is likely to differ from smaller, simpler models in terms of an exponentially increased number of theoretically possible routes through the process (Chapter 7).
- Even when the duration of reworked tasks is extremely low, project delays are likely due to the reordering of downstream tasks. Failure to correctly model either rework likelihood or task duration during rework can lead to poor predictions of project duration (Chapter 7 and 8).
- Processes which look similar from a task perspective can vary when examined from a product confidence perspective. The patterns of confidence growth indicate the likely design confidence variations for a given project deadline. Confidence data can help designers and managers quickly identify rework and hence minimise its impact on the process (Chapters 7 and 8).

9.2 Related research contributions

Design project planning is of critical importance to industry. In the current environment of increasing pressure to design better products to ever-tighter deadlines, there is a strong need to understand the resulting planning challenges and to support industry in overcoming the associated problems. En route to the above conclusions, which constitute a contribution of new knowledge in their own right, several additional research contributions were made to the area of design planning:

- Relevant research from different fields including complexity science, modelling, AI planning and operations research was reviewed (Chapter 3).
- A better understanding of planning practice was realised through an extensive industrial case study which also identified current planning challenges (Chapter 4).
- Task-precedences were introduced to the Signposting framework. This simplified the model-building process and allowed improved modelling of parallel tasks (Chapter 6).
- Extensions to the Signposting tool were realised to increase its practicality, provide support for modelling and visualisation and allow more in-depth process analyses. The extended Signposting tool can be used as a platform for future researchers (Chapter 6).
- A partially automated design process model generator was conceived and implemented. This tool provides academics interested in simulation analysis and visualisation of process models with quick and easy access to a vast array of process models which are otherwise difficult to obtain (Chapter 7).
- By leading to heuristics for planning, the simulation analyses demonstrated the utility of the model generator and the Signposting simulation tool (Chapters 7 and 8).

9.3 Contributions to Perkins

In addition to the more general contributions described above, this research also contributed to improved planning in Perkins. The explicit desire of Perkins to continue and extend its collaboration with the EDC, the company's decision to fund a summer student to work on a sub-topic of this thesis (relating Gantt charts and project plans) and their continuous commitment throughout this work all provide evidence of Perkins' satisfaction with the research. Specific contributions to Perkins are as follows:

- Perkins obtained an alternative perspective on important issues, which was free from the bias of internal company politics.
- Perkins' employees were made aware of relevant academic research which they would have probably overlooked otherwise. While they complained that

these ideas were sometimes overly theoretical, they were nonetheless keen to remain in touch with the latest academic research.

- Perkins gained insight into the merits and limitations of using confidence to plan and manage large-scale projects. The development of Perkins' confidence models took place in parallel to this research and it was possible to disseminate theoretical findings from academic research into the company.
- Perkins benefited from the definition of terminology that reconciled differences between task descriptions used in confidence plans and Gantt charts.
- Perkins was provided with heuristics for planning (Sections 7.6 and 8.5). These heuristics are particularly important in Perkins' current design context of increasing product and process risk and decreasing designer experience.

9.4 Evaluation against research objectives

In light of the contributions outlined above, the success of this work in addressing the research questions defined in Chapter 1 is considered here.

The first research question of this work concerns project planning in industry:

How are large-scale design projects planned and what are the challenges involved?

The industrial case study showed that designers, managers and plan administrators are all involved in creating plans and that these plans have multiple stakeholders, including purchasing, finance, sales and logistics personnel as well as core engineering staff. Also, while Gantt charts lie at the heart of planning at Perkins, several other documents are also used to plan projects. Despite the range of plans used, tacit knowledge from within the organisation is not captured and overview of experienced designers plays an important role in bridging the gap between project plans and reality. Overview also helps designers link together information from different plans, although some plan formats are better linked than others.

The case study also confirmed that the sensitivity of plans to different uncertainties is poorly understood. Perkins did not know which errors in the plan were likely to cause problems or which changes could lead to the greatest improvements in project

performance. Further, even when it was possible to identify high-risk tasks (such as miracle boxes) during the planning phase, none of the plan formats used in Perkins provided adequate visual representation of the associated project risks.

The second research question relates to sensitivity to process structure:

(How) Do process properties such as scale, connectivity-level and rework affect plannability?

Simulation analyses were performed to determine how scale, connectivity and rework characteristics are likely to affect project performance. Results show that even when process-specifics are unknown, inferences about likely process-behaviour can be made based on process structure.

This research showed why planning becomes increasingly difficult with increasing model-scale: the number of potential routes through the process grows exponentially. However, the normalised mean project duration remains comparatively stable. This implies that it is harder to predict the task-order for large-scale models but that good estimates for duration can nonetheless be expected. Simulation results also showed how project performance is likely to vary with increasing connectivity-level: both mean and standard deviation increase with rising connectivity level.

Numerous simulations into the effects of rework were undertaken. The most surprising result was that rework can delay processes considerably even when the time taken to rework tasks is negligible. This happens due to resulting changes to the order of downstream tasks. Results also showed how project duration is likely to vary with increasing rework-probability and that late discovery of rework almost inevitably leads to project delays. Process robustness can be increased by minimising rework, accelerating its discovery and through appropriate task-ordering of high-risk tasks.

Processes differ at a structural level in terms of scale, connectivity-level and rework-behaviour. The simulation results from this thesis show how these structural variations are likely to reflect themselves in terms of project performance. This empowers planners and managers to adapt their planning approach to fit the process structure and hence to create more robust plans.

The third research question concerns modelling the product-process link:

(How) Can modelling of product-process interdependencies improve the quality of plans and reduce process risks?

Conventional models of the design process, such as Gantt charts, fail to adequately account for the way in which technical issues are likely to affect the process and largely ignore the product-process link. This research demonstrated how confidence could be used to reflect product risks in the process domain, for example, confidence profiles can be used to quickly discover rework during process execution. It also showed how product confidence is likely to vary in the event of different task failures and in the context of tight deadlines (e.g. emissions legislation) which limits the time available for rework. This allows engineers and managers to develop plans which are more likely to result in high product confidence, subject to a given deadline. Finally, analyses showed how processes which look similar from a Gantt chart perspective can differ considerably from a confidence viewpoint and vice versa. It thus provides insights that facilitate management in making product-process trade-offs, and avoids solutions that are optimum in one domain but highly sub-optimal in the other, ultimately leading to better plans.

The fourth research question relates to the role of simulation in exploring the effects of different uncertainties:

How can simulation be most effectively deployed to determine the impact of different process properties on project performance and plannability?

This research used simulation analysis to examine 1) structural variation between processes and 2) process-sensitivity to different uncertainties. By better understanding the impact of different process-variations, it is possible to construct more robust plans. Simulations of both hypothetical models and real-world projects were performed, the former focusing on variations to process structure while the latter focused on project-specific uncertainties.

The use of hypothetical models overcomes concerns about inherent biases associated with real-world processes, thus allowing clearer focus on abstract properties. Results relating to rework, scale and connectivity-level showed how processes are likely to

behave based on their structural properties. Further, results from hypothetical models prepared the researcher for some of the modelling challenges which occur in industry, thus improving the efficiency of the latter exercise.

The simulation analysis of a real-world model showed which uncertainties have the greatest impact on project performance and under which circumstances. Understanding different sensitivities allows more focused model building, while also providing insights into which changes to make in order to improve performance or reduce risk.

A concern with much simulation work is validation. While it was not possible to completely validate all results obtained from simulation, reasonable agreement was observed between the simulation results and actual project events. While simulation analyses cannot be guaranteed to accurately predict all possible project behaviours, they nonetheless can provide useful insights into process behaviour, thus yielding appropriate strategies for risk-mitigation and robust planning. Overall, the research showed that simulation offers many potential benefits in understanding process behaviour and predicting and avoiding problems.

The final research question relates to tools and techniques for design project analysis:

How can support tools and techniques, which aim to support engineering-design planning, be improved?

Numerous tools already exist for process modelling and analysis. These tools, however, do not adequately capture rework, which is commonplace in design. They also fail to capture numerous other process risks and uncertainties and are limited in their ability to represent task-connectivity. The case study showed that despite these limitations, many projects succeed through the overview and experience of key personnel. As many of these senior employees retire against a backdrop of increasing product and process complexity, better tools for process modelling, analysis and planning are required.

Several enhancements to the Signposting tool were realised to meet this need. New visualisations (Gantt charts, DSMs, DMMs, scatter plots, histograms, and confidence profiles) were implemented in the Signposting tool to facilitate model-building and simulation analysis. Changes were implemented to allow better modelling of parallel

tasks, to allow the modelling of task-precedences and to improve the efficiency of the simulation algorithms. The utility of the improved tool was demonstrated through the simulation analysis of hypothetical and real-world models.

In addition to the research objectives defined in Chapter 1, more detailed requirements were defined in Chapters 5 and 6. All of these requirements were subsequently addressed in Chapters 6, 7 and 8.

9.5 Limitations of this work

Despite the success of this research in meeting its objectives, the generality of the findings, particularly those based on the simulation of the oil-and-cooling system model, is not proven. Analysis of additional industrial models would be required to confirm the generality of research findings in other contexts.

The analyses reported in this thesis suffer from the inherent limitations of all simulations. Simulation analyses are limited by the accuracy and detail of the underlying models – poor models will lead to poor results. Also, it is inherently difficult to validate simulation results concerning socio-technical systems (Johnson, 2001).

Although considerable effort has been expended on developing the Signposting tool and the model generator, user-testing of both tools by industrial practitioners has not been performed. While the approach taken by this thesis has been to perform process analyses externally and feed results back to the company in the form of heuristics for planning, an alternative approach would be to allow practitioners to apply the tool directly. This would require polishing and perfecting the software interfaces and evaluating the usability of the tool.

Another limitation of this work is that the heuristics developed in Chapters 7 and 8 have not been tested in industry. Although feedback from Perkins concerning the plausibility of these heuristics is positive, a thorough evaluation of their application would be needed to determine their impact in practice.

9.6 Opportunities for future work

Several opportunities exist for further work with a view towards improved engineering design process planning. This section begins by detailing specific ideas for developing the software tool that can be performed within the next year or so before proceeding to outline high-level, abstract opportunities for combining ideas from other research fields into the Signposting framework.

Integration with other Signposting research

In parallel to this research, other research within the EDC has focused on modelling and representation in the context of aero-engine design (Section 3.4.3). One obvious step, in transferring the theoretical knowledge from this research to industry, is the synthesis of both tools. In addition to providing benefits to industry, this approach would also result in a Signposting platform which could be further developed by other academics. However, while the alternative modelling and representation tool offers many benefits (Section 3.4.3), it is not currently capable of modelling confidence in a manner compatible to this research.

Research into strategies for picking tasks

Chapter 6 of this thesis describes how different algorithms have been implemented in the simulation tool to better reflect the way in which tasks are chosen in industry. Simulation analyses to compare the effect of the different approaches to picking tasks, followed by industrial evaluation of results could lead to heuristics for task-choice in different contexts such as:

- pick the task with the highest output state;
- pick the task which is most likely to deliver a satisfactory outcome;
- pick tasks based on the average output state;
- pick tasks based on a combination of output state, time, cost and resource availability.

Further development of the modelling and simulation software

Increased functionality could be added to the Signposting tool to allow better representation of parameter connectivity within design process models. Likewise, representations of resource profiles could be implemented and results compared against confidence, cost and time data to allow more sophisticated process analysis.

Algorithms could be developed to optimise process models based on different objective functions such as time, cost, confidence and resource utilisation.

Work is also ongoing to develop user-friendly interfaces for the Signposting tool. The usability of these interfaces should also be tested in an industrial setting as a prerequisite to transforming the current proof-of-concept software implementation to a more professional tool, suitable for direct application in industrial practice.

Further analyses into process structure

In addition to variations in connectivity level, process structure can vary in terms of connectivity patterns. For example, some processes have clusters of highly connected tasks while other tasks are largely unconnected. It would be interesting to generate hypothetical models which vary in terms of the connectivity patterns, rather than solely the degree of connectivity.

Likewise, the nature of rework in processes differs not only in terms of task failure probability and task rework duration but also in terms of the number of tasks which must be reworked and the timing of task failures. Analyses to explore the impact of different degrees of rework (number of tasks affected and the size of iterations) and timing of task failures would be useful. Initial simulation results indicate that task failures which occur late in the process are difficult to recover from, while early failures are likely to lead to alternative process routes.

Several additional simulation analyses could also be performed on both hypothetical and real-world models. For example, it would be interesting to explore process sensitivity to:

- Different degrees of duration uncertainty;
- Different distributions of duration uncertainty;
- Resource variations;
- Uncertainty in requirements (goal state);
- Variations in task confidence contributions;
- Multiple factors in combination.

The final topic, multi-variant simulation analyses based on the factors already considered in this thesis (i.e. scale, connectivity level, rework probability, and rework duration), warrants further discussion. Such analyses could shed light on the interaction between these variables. For example, it would be interesting to explore how simultaneously varying rework probability and model scale would affect project

duration and to consider whether resulting observations are robust for all levels of connectivity. It is worth bearing in mind, however, that the number of model-variants required would be considerable due to the number of factors and factor-variations necessary for such analyses and appropriate experimental design would be essential to the success of such work. Also, validation of results would likely prove problematic. In addition to multi-variant analyses, more extensive analyses relating to the above factors would also be possible. For example, simulation analyses could be used to explore the effect of variations in connectivity level or rework duration for different task clusters within a single model (the analyses in Chapter 7 typically assume uniform connectivity and rework duration throughout the models). Likewise, the effects of task-reordering and omission during rework cycles could be explored. Another important topic, which could be explored through the simulation analysis of hypothetical models, is the use of gateways; this topic is discussed in the following section.

Using hypothetical models to plan staged gateways

Staged gateway models, such as the Perkins NPI process described in Section 4.3.1, are commonly used in industry (Cooper, 1994). While the utility of staged gateways is widely accepted, deciding on the timing and scope of gateways within a process is not always straightforward.

Typically, gateways are aligned to phases in the design process such as product and process design or product and process validation. Unfortunately, the distinction between these phases is not always clear, particularly in the context of design iteration where unexpected results during validation may lead to redesign work. Also, gateways are usually defined in terms of the tasks which must be completed while they are intended to reflect the maturity of different design parameters (e.g. components) as the process proceeds.

This leads to problems when tasks fail to deliver the expected confidence – officially the gateway is reached but, unofficially, the confidence gap signifies a downstream problem. In the worst case scenario, poorly timed gateways may actually encourage stakeholders to exaggerate the level of task completion by placing them under excessive time pressure. The simulation analysis of hypothetical models would highlight discrepancies between task-based and confidence-based gateways, thereby helping to overcome this problem.

Even in the absence of confidence data, however, simulation analyses into rework would be useful in highlighting iteration loops within the process and hence providing guidance on appropriate gateway timing. For example, it may make sense to perform several high-risk, independent tasks within a single stage in order to localise the risk and achieve multiple delays in parallel, hence minimising the overall project delay.

In complement, simulation analyses into scale and connectivity could provide insight into the most suitable number of stages for different processes. The results from this thesis indicate that the number of possible process routes depends on scale and connectivity. Gateways have the effect of resetting the process to a predefined state and thereby counteracting the undesired complexity associated with rising scale and connectivity. Further simulation analyses could inform decisions on the gateway strategies by reflecting the process complexity in terms of scale, connectivity and rework.

Engineering change support from Signposting

Engineering change management is a major challenge in industry (Clark and Fujimoto, 1991; Eckert et al., 2004b; Flanagan et al., 2003b; Jarratt, 2004). A change to one component can lead to changes in other dependent components, driving rework in the associated tasks and often leading to project delays. Ongoing research in the EDC concerns the development of tools to support industrial change management (Keller et al., 2005).

Resulting product models contain rich data about the way in which different components are connected and how changes are likely to propagate. By linking this product data to tasks in design processes, detailed models of likely process rework resulting from engineering changes could be created. While interlinking change and Signposting tools would prove challenging, such research would provide benefits to academia and industry alike.

Incorporation of planning and scheduling techniques

As discussed in the literature review, extensive research has been performed in the field of AI planning and scheduling. Although this research is often focused on tightly defined problems, opportunities exist to incorporate concepts from AI planning into the Signposting framework.

Agent-based systems are also popular in the planning community. Maes (1995, pp 108) provides the following definition “Autonomous agents are computational

systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed.” Although Signposting was not originally conceived as an agent-based system, taking an agent perspective on Signposting tasks may prove useful. By considering Signposting activities as agents capable of autonomously sensing their environment, extensive research into agent based planning could be coupled with the existing Signposting tool.

Another concept under development in the planning and scheduling community is the development of optimisation heuristics based on patterns in plans (Spalazzi, 2001). Similar patterns in other plans, or other parts of the same plan, are identified and optimisation based on different heuristics is attempted. While this approach may not be easily adapted to design process planning due to its highly uncertain nature, the topic is nonetheless a worthy candidate for further research.

Communication support from confidence profiles

Communication in engineering design is complex and resulting problems are a major concern in industry (Flanagan et al., 2003a). Many designers do not know how their own tasks fit into the overall product design and current process models do not provide the designer with sufficient information at a parameter level.

The application of Signposting to communication support could address some of these concerns. Signposting models are very rich in data; in contrast to the other models, they contain sufficient information to visually represent information flow in the form of confidence profiles. In addition to applications in planning, confidence profiles show where parameter-specific information is coming from and going to. Hence, they could be used to help designers follow up information and engage in negotiations with appropriate members of staff, ensuring that others are not unnecessarily bothered. Opportunities for future work include providing enhanced computer support and increasing the amount of information contained in confidence profiles by using annotations.

9.7 Summary

Through the simulation of both real-world and hypothetical models, this thesis demonstrated the utility of an enhanced Signposting tool in dealing with design planning challenges. Industrial feedback, concerning the utility of the research, has

been highly positive. Although this work will neither save the Vasa, nor solve the current problems of NASA (Nelson, 2006), it nonetheless provides valuable insights into issues that have troubled designers for centuries.

Appendix

This appendix describes the format of the models analysed using the Signposting tool and visually represents how these models behave during simulation. Descriptive comments are shown in blue.

Example Hypothetical Model

```

2 /*The format of the model – 2 indicates that the model includes both confidence and precedence
   parameters, 1 indicates that cumulative confidence parameters are not included*/
1 //The number of resources
Example_RESOURCE RETURNABLE 5.0 /*Name, type and number of resource. Returnable
   indicates that the resource is returned to the resource pool once the task finishes, e.g. human
   resources. USES_UP indicates that the resource gets used up by the task, e.g. material. */
56 //The number of parameters
From Task_0 to Task_2 //Name of parameter 0
From Task_0 to Task_4 //Name of parameter 1
From Task_1 to Task_6 //Name of parameter 2
From Task_0 to Task_8 //Name of parameter 3
From Task_6 to Task_8 //Name of parameter 4
From Task_3 to Task_11 //Name of parameter 5
From Task_4 to Task_11 //Name of parameter 6
From Task_5 to Task_11 //Name of parameter 7
From Task_9 to Task_11 //Name of parameter 8
From Task_4 to Task_12 //Name of parameter 9
....
ComponentConf //Name of parameter n
LateConfGrowth //Model name
2 1000.0 900.0 2000.0 1800.0 /*Number of goal states defined (2), target cost (1000),
   target duration (900), max cost (2000), max duration (1800). Different goal states may be defined to
   reflect intermediate goals within the process, e.g. gateways. For this example, only the initial state
   (State0) and the final state (State1) are defined. All goal states are defined in terms of the parameter
   values*/
State0 (0 0) (1 0) (2 0) (3 0) (4 0) (5 0) (6 0) (7 0) (8 0) (9 0) (10 0) (11 0) (12 0) (13 0) (14 0) (15 0)...
State1 (0 1) (1 1) (2 1) (3 1) (4 1) (5 1) (6 1) (7 1) (8 1) (9 1) (10 1) (11 1) (12 1) (13 1) (14 1) (15 1)...
20 //Number of tasks
Task_0 G0 //Task name(Task0), task group (G0). Similar tasks are allocated to the same task group
InputState /*The input state is usually specified in terms of the parameter confidences. The blank entry
   in this case indicates that no confidence in any parameter is required and hence that Task_0 is not
   dependent on any previous tasks. A more typical input state is shown for Task_1 below*/
Resource (0 1.0) //The task requires 1 unit of resource 0
ResourceVar (0 0.0) //There is no resource variance associated with this task
1 //The task has a single outcome
1.0 //The probability of this outcome taking place is 1
OutputState0 (0 1) (1 1) (3 1) (15 1) (19 1) (32 1) (35 1) /*The output state is defined in terms of
   parameter values. This output state indicates that parameters 0, 1, 3, 15, 19, 32 and 35 reach a value
   of 1 as a result of Task_0. In addition, parameter 55 reaches a confidence value of 4.*/
ConfCondition (55 0.0 4.0) /*If the confidence associated with parameter 55 is at least 0.0 when the
   task commences, then this parameter confidence should be increased by 4.0 */
CostMean (0 5.0) //Mean cost for Task_0
CostVar (0 0.0) //Cost variance for Task_0
DurationMean (0 5.0) //Mean duration for Task_0
DurationVar (0 0.0) //Duration variance for Task_0
Double_1 (0 0.0) //A variable of type double which can be used to store additional task information
Double_2 (0 0.0) //A variable of type double which can be used to store additional task information
/*Double_1 and Double_2 were used respectively to store information about cost uncertainty and
   duration uncertainty in a previous Signposting implementation. Such information is not used by the
   simulation engine for this thesis. The associated fields in the input file are sometimes overwritten to
   store additional information on task properties, e.g. task duration during rework expressed as a
   percentage of the original task duration*/

```


Task_1 G1

InputState (0 1) /*The state (0 1) indicates that a value of 1 in parameter zero is required and hence that Task_1 is dependent on Task_0, which creates this information*/

Resource (0 1.0)

ResourceVar (0 0.0)

2 //The task has two possible outcomes

0.8 //The probability of this outcome taking place is 0.8

OutputState0 (2 1) (20 1) (36 1) //Output state 0 affects precedence parameters 2, 20 and 36

ConfCondition (55 0.0 6.0) //Output state 0 increases confidence parameter 55 by 6

0.2 //The probability of this outcome taking place is 0.2

OutputState1 (0 0) //Output state 1 causes the parameter 0 to be reset thus driving rework of Task_0

ConfCondition (55 0.0 -4.0) //Output state 0 decreases confidence parameter 55 by 4

CostMean (0 9.0) (1 8.0) //Cost mean for both outcomes (mappings)

CostVar (0 0.0) (1 0.0) //Cost variance for both outcomes

DurationMean (0 9.0) (0 6.0) //Duration mean for both mappings

DurationVar (0 0.0) (0 0.0) //Duration variance for both mappings

Double_1 (0 0.0) (0 0.0) //Cost uncertainty for both mappings

Double_2 (0 0.0) (0 0.0) //Duration uncertainty for both mappings

... //Additional tasks follow the same structure

Simplified Example Model (to demonstrate simulation)

2

1

Example_RESOURCE RETURNABLE 2.0

11

P0

P1

P2

P3

P4

P5

P6

P7

P8

P9

C

ExampleModel

2 1000.0 1000.0 1000.0 1000.0

ConfState0 (10 20)

State0 (0 0) (1 0) (2 0) (3 0) (4 0) (5 0) (6 0) (7 0) (8 0) (9 0)

State1 (0 1) (1 1) (2 1) (3 1) (4 1) (5 1) (6 1) (7 1) (8 1) (9 1)

10

T0 G0

InputState

Resource (0 1.0)

ResourceVar (0 0.0)

1

1.0

OutputState0 (0 1)

ConfCondition (10 0.0 5.0)

CostMean (0 2.0)

CostVar (0 0.0)

DurationMean (0 3.0)

DurationVar (0 0.0)

T0_Double1 (0 1.0)

T0_Double2 (0 0.0)

T1 G1

InputState

Resource (0 1.0)

ResourceVar (0 0.0)

1

1.0

OutputState0 (1 1)

ConfCondition (10 0.0 4.0)

CostMean (0 7.0)

CostVar (0 0.0)

DurationMean (0 5.0)

DurationVar (0 0.0)

T1_Double1 (0 1.0)

T1_Double2 (0 0.0)

T2 G2

InputState (0 1) (1 1)

Resource (0 1.0)

ResourceVar (0 0.0)

1

1.0

OutputState0 (2 1)

ConfCondition (10 0.0 3.0)

CostMean (0 4.0)

CostVar (0 0.0)

DurationMean (0 5.0)

DurationVar (0 0.0)

T2_Double1 (0 1.0)

T2_Double2 (0 0.0)

T3 G3

InputState

Resource (0 1.0)

ResourceVar (0 0.0)

1

1.0

OutputState0 (3 1)

ConfCondition (10 0.0 10.0)

CostMean (0 7.0)

CostVar (0 0.0)

DurationMean (0 5.0)

DurationVar (0 0.0)

T3_Double1 (0 1.0)

T3_Double2 (0 0.0)

T4 G4

InputState (1 1) (3 1)

Resource (0 1.0)

ResourceVar (0 0.0)

1

1.0

OutputState0 (4 1)

ConfCondition (10 0.0 5.0)

CostMean (0 2.0)

CostVar (0 0.0)

DurationMean (0 3.0)

DurationVar (0 0.0)

T4_Double1 (0 1.0)

T4_Double2 (0 0.0)

T5 G5

InputState (2 1) (4 1)

Resource (0 1.0)

ResourceVar (0 0.0)

1

1.0

OutputState0 (5 1)

ConfCondition (10 0.0 2.0)

CostMean (0 4.0)

CostVar (0 0.0)

DurationMean (0 3.0)

DurationVar (0 0.0)

T5_Double1 (0 1.0)

T5_Double2 (0 0.0)

T6 G6

InputState (3 1) (4 1)

Resource (0 1.0)

ResourceVar (0 0.0)
1
1.0
OutputState0 (6 1)
ConfCondition (10 0.0 5.0)
CostMean (0 9.0)
CostVar (0 0.0)
DurationMean (0 5.0)
DurationVar (0 0.0)
T6_Double1 (0 1.0)
T6_Double2 (0 0.0)

T7 G7
InputState (3 1) (5 1) (6 1)
Resource (0 1.0)
ResourceVar (0 0.0)
2
0.8
OutputState0 (7 1)
ConfCondition (10 0.0 3.0)
0.2
OutputState1 (3 0) (5 0) (6 0)
ConfCondition (10 0.0 -17.0)
CostMean (0 5.0) (0 3.0)
CostVar (0 0.0) (0 0.0)
DurationMean (0 3.0) (0 3.0)
DurationVar (0 0.0) (0 0.0)
T7_Double1 (0 1.0) (0 1.0)
T7_Double2 (0 0.0) (0 0.0)

T8 G8
InputState (7 1)
Resource (0 1.0)
ResourceVar (0 0.0)
1
1.0
OutputState0 (8 1)
ConfCondition (10 0.0 0.0)
CostMean (0 2.0)
CostVar (0 0.0)
DurationMean (0 3.0)
DurationVar (0 0.0)
T8_Double1 (0 1.0)
T8_Double2 (0 0.0)

T9 G9
InputState (7 1)
Resource (0 1.0)
ResourceVar (0 0.0)
1
1.0
OutputState0 (9 1)
ConfCondition (10 0.0 6.0)
CostMean (0 5.0)
CostVar (0 0.0)
DurationMean (0 4.0)
DurationVar (0 0.0)
T9_Double1 (0 1.0)
T9_Double2 (0 0.0)

When the above model is simulated in the Signposting tool, more than one possible route through the process is possible based on the task precedences and resource constraints defined. For example, T0, T1 or T3 are all possible and useful at the start of the process but only two of these tasks can be executed in parallel due to resource constraints. Likewise, the failure of T7 is probabilistic and will only occur in 20% of cases, thus leading to different process routes. One possible route is shown below (Fig. A.1). As each task is executed, the current state of the process is updated to reflect the information created in terms of the precedence parameters, P0 to P9 and the confidence parameter C. The process duration (D) and the accumulated cost (£) are also updated as the simulation progresses.

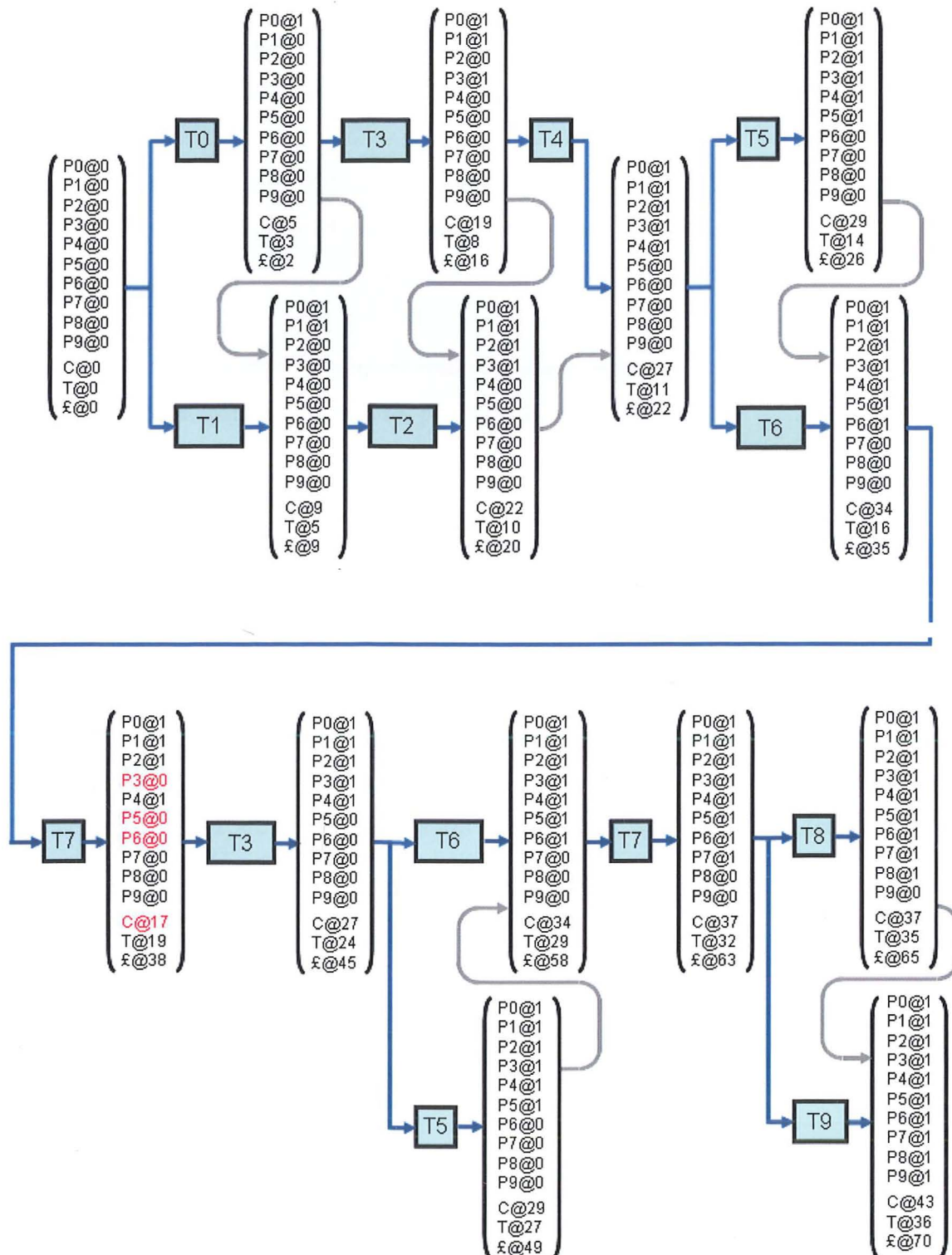


Fig. A.1 A visual representation of simulation code execution

Bibliography

Agar, M. (1996) *The professional stranger: an informal introduction to ethnography*. Academic Press, New York

Allen, R. (2000) *Workflow: An introduction in The Workflow Handbook 2000*, Layna Fischer(ed), Workflow Management Coalition (WfMC).

Austin, S., Steele, J., MacMillan, S., Kirby, P. and Spence, R. (2001) Mapping the conceptual design activity of interdisciplinary teams. *Design Studies*. Vol. 22(3). pp.211-232

Bacharach, S.B. (1989) Organizational theories: some criteria for evaluation. *Academy of Management Review* 14, pp. 496-515.

Barrett, A., and Weld, D. (1994) Partial-order planning: Evaluating possible efficiency gains. *Artificial Intelligence*, 67 (1), 71 {112}.

Bichlmaier, C. (2000) *Methoden zur flexiblen Gestaltung von integrierten Entwicklungsprozessen*. München, Dissertation, TU München

Biller, B. and Nelson, B.L. (2002) Answers to the top ten input-modeling questions. *Proceedings of the 2002 Winter Simulation Conference*. Piscataway, New Jersey.

Blessing, L.T.M. (1994) *A Process-Based Approach to Computer Supported Engineering Design*, PhD thesis. Twente, Netherlands.

Blessing, L.T.M., Chakrabarti, A. and Wallace, K.M. (1995) A design research methodology. *ICED 95*. Prague, Czech Republic, pp.50-55

Bracewell, R.H., Shea, K., Langdon, P.M., Blessing, L.T.M. and Clarkson, P.J. (2001) A methodology for computational design tool research. *ICED 01*. Glasgow, UK

Breakwell, G., Hammond, S., Fife-Schaw, C. (2000) *Research Methods in Psychology*, Sage Publications

Browning, T.R. (1998) *Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development* Ph.D. Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA

Browning, T.R. (2001) Applying the Design Structure Matrix system to decomposition and integration problems: A review and new directions. *IEEE Transactions on Engineering Management*. Vol.48. pp.292-306.

Browning, T.R. (2004) Design planning and modelling, in Clarkson, P.J. and Eckert, C.M. (eds.), *Design process improvement - a review of current practice*, Springer

Browning, T.R. and Ramasesh, R.V. (2005) *Modeling the Product Development Process: A Survey of the Literature* TCU M.J. Neeley School of Business Working Paper

Bryman, A. (2001) *Social Research Methods*. New York: Oxford University Press.

Bucciarelli, L.L. (1994) *Designing Engineers*. Cambridge: MIT Press

Byrne, D.S. (1998) *Complexity theory and the social sciences: an introduction*. London: Routledge.

Callan, R. (2003) *Artificial Intelligence*, Palgrave Macmillan

Cantamessa, M. (2001) Design research in perspective – a meta-research on ICED 97 and ICED 99. ICED01. Glasgow. UK. pp.29-36.

Carrascosa, M., Eppinger, A. and Whitney, D. (1998) Using the designs structure matrix to estimate product development time. *Proc. ASME Des. Eng. Tech. Conf. (Design Automation Conf.)*, Atlanta, GA

Carter, M.W. and Price C.C. (2001) Operations Research. A Practical Introduction, CRC Press, Boca Raton, FL.

Charniak, E. and McDermott, D. (1987) Introduction to Artificial Intelligence. Addison Wesley: Canada

Checkland, P. (1981) Systems Thinking, Systems Practice, Chichester, John Wiley.

Chwif, L., Barretto, M.R., and Paul, R.J. (2000) On Simulation Model Complexity, Proceedings of the 2000 Winter Simulation Conference, Florida, USA

Cho, S.H. and Eppinger, S.D (2005) A Simulation-Based Process Model for Managing Complex Design Projects. IEEE Transactions on Engineering Management, Vol.52(3)

Clark, K.B. and Fujimoto, T. (1991) Product development performance: strategy, organization and management in the world auto industry. HBS Press

Clarkson, P.J. and Eckert, C.M. (2004) Design process improvement - a review of current practice, Springer

Clarkson P.J. and Hamilton J.R. (2000) Signposting: a parameter-driven task-based model of the design process, Research in Engineering Design, pp.18-38

Clarkson, P.J., Melo, A.F. and Eckert, C.M. (2000a) Visualization of routes in design process planning. Information Visualisation (IV), pp.155-164

Clarkson, P.J., Melo, A.F. and Connor, A.M. (2000b) Signposting for design process improvement, a dynamic approach to design process planning. Artificial Intelligence in Design (AID 2000), pp.333-353

Cooper, K.G. (1993) The Rework Cycle: Why Projects Are Mismanaged, PMNetwork, February, pp.5-7

Cooper, R.G. (1994) Third-generation new product processes, *Journal of Product Innovation Management*. 11, pp.3-14.

Cross, N. (1989) *Engineering Design Methods*, John Wiley and Sons, Chichester.

CTG (1997) *An Introduction to Workflow Management Systems. Models for Action Project: Developing Practical Approaches to Electronic Records Management and Preservation*, Center for Technology in Government, University at Albany

Danilovic, M and Browning, T.R. (2004), *A Formal Approach for Domain Mapping Matrices (DMM) to complement Design Structuring Matrices (DSM)*, DSM workshop, Cambridge, UK

Dick, B. (1999) What is action research? Available on line at <http://www.scu.edu.au/schools/gcm/ar/whatisar.html>

Dick, B. and Swepson, P. (1997) *Action research FAQ: frequently asked questions* <http://www.scu.edu.au/schools/gcm/ar/arp/arfaq.html>

Dou, W.C. and Cai, S.J. (2002) Hierarchical modeling and decomposition for workflow system based on Petri Net, *Advances in Concurrent Engineering: Proceedings of Concurrent Engineering Conference 2002*, Cranfield University. pp.717-728

Doyle, P. (2005) <http://www.cs.dartmouth.edu/~brd/Teaching/AI/Lectures/Summaries/planning.html>

DTI (1992) *Successful Product Development*, Department of Trade and Industry, UK

Dubin, R. (1969) *Theory building*. New York: The Free Press

Dym, C.L. (1994) *Engineering Design: A Synthesis of Views*, Cambridge University Press, Cambridge, UK

Earl, C., Johnson, J. and Eckert, C.M. (2004) Complexity in design, in Design Process Improvement, Clarkson and Eckert (eds), Springer

Eckert, C.M. and Clarkson, P.J. (2003) The reality of design process planning, in ICED03, International Conference on Engineering Design, Stockholm, Sweden.

Eckert, C.M., Clarkson, P.J. and Earl, C. (2005) Predictability of change in engineering: a complexity view, ASME DTM 2005. 2005, Long Beach, California, USA

Eckert, C.M., Clarkson, P.J. and Stacey, M. (2003) The spiral of applied research: a methodological view on integrated design research. ICED03. Stockholm, Sweden

Eckert, C.M., Stacey, M.K. and Clarkson, P.J. (2004a) The lure of the measurable in design research. Design 04, Dubrovnik, Croatia, 21-26

Eckert, C.M., Clarkson, P.J. and Zanker, W. (2004b) Change and Customisation in Complex Engineering Domains, Research in Engineering Design, Vol.15(1), pp.1-21

Eckes, G. (2001) The six sigma revolution: how General Electric and others turned process into profits. Published: New York; Chichester: John Wiley, 2001.

Eppinger, S.E. (2001) Innovation at the Speed of Information, Harvard Business Review, vol. 79(1), pp.149-158

Evalife, (2005) The domain between order and chaos, Online article, http://www.evalife.dk/projects/showproject.php?file=soc_operators

Evans, J.H. (1959) Basic Design Concepts, Naval Engineers Journal, November.

Evans, M. (2005) Overdue and over budget, over and over again, The Economist, June 9th, http://www.economist.com/business/displayStory.cfm?story_id=4065653

Fairley R. and Willshire, M. (2003) Why the Vasa Sank: 10 Problems and Some Antidotes for Software Projects, IEEE Software Vol.20(2). pp.18-25

Feldman, D.P. and Crutchfield, J. (1998) A Survey of Complexity Measures, The Santa Fe Institute

Fisher, B., Agelidis, M., Dill, J., Tan P., Collaud, G. and Jones, C. (1997), CZWeb: Fish-eye Views for Visualizing the World-Wide-Web in Proceedings of HCI International '97. p. 719-722.

Flanagan, T., Eckert, C.M. and Clarkson, P.J. (2003a) Parameter trails. ICED03. Stockholm, Sweden, pp.71-73

Flanagan, T., Eckert, C.M., Eger, T., Smith, J. and Clarkson, P.J. (2003b) A functional analysis of change propagation. ICED 03. Stockholm, Sweden, pp.441-442

Flanagan, T., Eckert, C.M., Keller, R. and Clarkson, P.J. (2005a) Robust scheduling of design tasks using simulation. ICED 05, Melbourne, Australia, pp.365-366

Flanagan, T., Eckert, C.M., Keller, R. and Clarkson, P.J. (2005b) Rework revisited: the criticality of iteration due to task failure in NPD planning, IEEE International Engineering Management Conference (IEMC 05), St. Johns, Newfoundland, Canada

Flanagan, T., Keller, R., Eckert, C.M. and Clarkson, P.J. (2006) Bridging the gaps between project plans and reality: the role of overview. TMCE 2006. Ljubljana, Slovenia

Flattery, M. (2005) Workflow systems, Tessella Support Services Technical Report

Ford, D. and Serman, J. (2003) The Liars Club: Concealing Rework in Concurrent Development, Concurrent Engineering, Vol.11, pp211-219

Gass, S.I. and Assad, A.A. (2005) An annotated timeline of Operations Research, an informal history, Kluwer

Gell-Mann, M. (1995) What is Complexity? Complexity. Vol.1(1). John Wiley and Sons Inc.

Greenwald, J. (2001) Inside the Ford/Firestone Fight, Time Magazine Online, May 2001, <http://www.time.com/time/business/article/0,8599,128198,00.html>

Haga, W.A and O'Keefe, T. (2001) Crashing PERT networks: a simulation approach, 4th International conference of the Academy of Business and Administrative Sciences Conference, Quebec, Canada

Hamilton, J.R. (1999) The capture and representation of knowledge to support adaptive aerospace design, Ph.D. Thesis, Cambridge University Engineering Department

Hammersley, M. and Atkinson, P. (1995) Ethnography: Principles in Practice, London, Routledge (first published 1983)

Henderson, K. (1999) On Line and On Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering, Cambridge: MIT Press

Hendrickson, E. (2002) Going down with the ship, SQA Newsletter, Volume 14, Issue 4 ISSN 1325-2070

Hillier, F.S. and Lieberman G.J. (1974) Operations Research. 2nd ed. San Francisco: Holden-Day

Horowitz, J. (1967) Critical path scheduling: management control through CPM and PERT, Krieger Publishing, Malabar Florida, ISBN: 0898740886

Horvath, I., Vergeest, J.S.M. and van der Vegte, W. (2000) Modeling design processes and designer decisions with advanced Petri-nets, Microcad 2000, Miskolc, Hungary

Ingalls, R.G. (2002) Introduction to simulation, Winter Simulation Conference, 7-16, Arlington, Virginia

Irving, T. (1999) The Tacoma Narrows Bridge failure, Vibrationdata Technical Report, <http://www.vibrationdata.com/Tacoma.htm>

Jarratt, T.A.W., Eckert C.M., Weeks, R. and Clarkson, P.J. (2003) Environmental legislation as a driver of design, ICED03. Stockholm, Sweden

Jarratt, T.A.W. (2004) A model-based approach to support the management of engineering change, PhD-thesis, Cambridge University Engineering Department

Jarrett, J.P. (2000) Technology or methodology? An approach to designing better turbomachinery. PhD-thesis. Cambridge University Engineering Department

Jarrett, J.P. and Clarkson, P.J. (2001) Improving the product by accelerating the process, ICED01. Glasgow, Scotland.

Johnson, J. (1995) The multidimensional networks of complex systems, in Casti, J., Thord, R. and Batten, D. (eds) Networks in Action, Springer

Johnson, J. (2001) The Can You Trust It? Problem of Simulation Science in the Design of Socio-technical Systems, Complexity, Vol. 6(2). pp.34-40.

Johnson, J. H. (1983) Hierarchical Set Definition by Q-analysis, Part II. Traffic on the Hierarchical Backcloth. International Journal of Man-Machine Studies 18: 467-487.

Juntunen, A. (2001) Professional and lay care in the Tanzanian village of Ilembula, PhD Thesis, Oulu University Press, <http://herkules.oulu.fi/isbn9514264312/isbn9514264312.pdf>

Keller, R., Eckert, C.M. and Clarkson, P.J. (2005) Multiple views to support engineering change management for complex products. 3rd International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV 05), London, UK.

Keller, R., Flanagan, T., Eckert, C.M. and Clarkson, P.J. (2006) Two sides of the story: Visualising Products and Processes in Engineering Design, 10th International Conference Information Visualisation, July 2006, London (accepted)

Kerzner, H. (1992) Project management: a systems approach to planning, scheduling, and controlling, 4th ed. ISBN: 0442010850, Van Nostrand Reinhold, New York

Kettle, B.P., Hendler, J.A., Andersen, W.A., and Evett, M.P. (1994) Massively parallel support for case-based planning. IEEE Expert, Vol 2., pp.8-14

Kim, C.H., Yim, D. and Weston, R.H. (2001) An integrated use of IDEF0, IDEF3 and Petri-net methods in support of business process modelling. Proc. of the Institution Mechanical Engineers, Journal of Engineering Manufacture, 215, pp.317-329

Lions, J.L. (1996) ARIANE 5 Flight 501 Failure, Report by the Inquiry Board Paris, <http://ravel.esrin.esa.it/docs/esa-x-1819eng.pdf>

Maes, P. (1995) Artificial Life Meets Entertainment: Life like Autonomous Agents, Communications of the ACM, 38, 11, 108-114

Malcolm, D.G., Roseboom, J.H., Clark, C.E. and Fazar, W. (1959) Applications of a Technique for Research and Development Program Evaluation (PERT). Operations Research. Vol.7(5)

Marca, D.A. and McGowan, C.L. (1993) IDEF0 - SADT Business Process and Enterprise Modelling, EMcGraw-Hill

McMahon, C.A., Sims Williams, J.H. and Brown, K.N. (1993) A transformation model for the integration of design computing, ICED93. The Hague, Netherlands, pp.1586-1593

Melo, A.F. and Clarkson, P.J. (2001) Design process planning using a state-action model, ICED01. Glasgow, Scotland

Melo, A.F. (2002) A state-action model for design process planning, PhD-thesis, Cambridge University Engineering Department

Meyer, R.J., Menzel, C.P., Painter, M.K., deWitte, P.S., Blinn, T., and Perakath, B. (1995) Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report, Knowledge Based Systems Incorporated, Technical Report

Mitleton-Kelly, E. (2000) Complexity: Partial Support for BPR? Systems Engineering for Business Process Change, Springer-Verlag, Chapter 3

Minneman, S.L. (1991) The Social Construction of a Technical Reality: Empirical Studies of Group Engineering Design Practice. PhD Thesis, Department of Mechanical Engineering, Stanford University, Stanford, CA

Morrison, J.M. (2003) A contribution to scientific knowledge. Working paper. University of Stellenbosch Business School

Nance, R.E. and Sargent, R.G. (2002) Perspectives on the Evolution of Simulation, Operations Research, 50(1), pp.161-172.

National Audit Office (2004) Ministry of Defence: Major Projects Report 2003, http://www.nao.org.uk/publications/nao_reports/03-04/0304195.pdf

Nelson, B.L. (2004) 50th Anniversary Article: Stochastic Simulation Research in Management Science, Management Science 50, pp.855-868

Nelson, D.A. (2006) The vision is collapsing. www.nasaproblems.com

Newell, A. and Simon, H. (1963) GPS, a Program that Simulates Human Thought, in Computers and Thought, E. Feigenbaum and J. Feldman (eds.) pp.279-293. McGraw Hill

Nichols, K. (1990) Getting engineering changes under control, J. of Engineering Design, Vol.1(1)

Noran, O.S. (2000) Business modelling: UML vs. IDEF, School of CIT, Griffith University, Australia

O'Donovan, B. (2004) Modelling and simulation of engineering design processes, PhD-thesis, Cambridge University Engineering Department

O'Donovan, B., Eckert, C.M. and Clarkson, P.J. (2004) Simulating design processes to assist in design process planning in ASME DETC 2004, Salt Lake City, Utah, USA

Oppenheim, A. (2000) Questionnaire design, interviewing and attitude measurement, Pinter Publishers

Packham, I.S.J. and Denham, S.L. (2003) Visualisation Methods for Supporting the Exploration of High Dimensional Problem Spaces in Engineering Design, International Conference on Coordinated Multiple Views in Exploratory Visualisation, London, UK, IEEE Computer Society.

Pahl, G. and Beitz, W. (1996) Engineering Design, 2nd ed., trans. Wallace K.M., Blessing, L.T.M. and Bauert, F., Springer, London.

Patton, M.Q., (1990) Qualitative evaluation and research methods, (Second Edition). Newbury Park, CA: Sage Publications.

Pidd, M. (1992) Computer simulation in management science, Third Edition, ISBN: 0-471-93462-3, Wiley

Pritsker, A. and Happ, W.W. (1966) GERT: Graphical evaluation and review technique, Part I, Fundamentals. Journal of Industrial Engineering, 17(5), pp.267-274.

Pritsker, A. and Sigal, C.E. (1983) Management Decision Making: A Network Simulation Approach. Prentice Hall Inc., Englewood Cliffs, NJ

PMI 2000 (2000) A guide to the Project Management Body of Knowledge, www.pmi.org

Prigogine, I. (1997) The end of certainty: time, chaos, and the new laws of nature, New York; London: Free Press

Rittel, H. and Webber, M.M. (1984) Planning Problems are Wicked Problems. In N. Cross (Ed.), Developments in Design Methodology, John Wiley & Sons, New York, pp.135-144.

Rizzoli, A.E. (2005) <http://www.idsia.ch/~andrea/simtools.html>

Ruppert, P.E. (1994) CFD and the changing world of Airplane Design, AIAA Wright Brothers Lecture, AIAA Press

Rubin, H.J. and Rubin, I.S. (1995) The art of hearing data, Sage Publications.

Russel, S. and Norvig, P. (2003) Artificial Intelligence: A modern approach. Second edition, Prentice Hall

Sargent, R.G. (1998) Verification and validation of simulation models, Proceedings of the 30th conference on Winter simulation, Washington, D.C.

Shah, J.J. (1996) Research opportunities in Engineering Design, National Science Foundation, Strategic Planning Workshop Final Report, Arizona, USA

Shigley, J.E. and Mischke, C.R. (1989) Mechanical Engineering Design, McGraw-Hill

Silverman, D. (ed) (2004) Qualitative Research. Theory, Method and Practice. Second Edition. Sage Publications

Simons, C.S. (2000) Change propagation in product design, MPhil-thesis, Cambridge University Engineering Department

Simpson, J., and Weiner, E.C. (Eds) (1989) Oxford English Dictionary, (Second Edition) Clarendon Press, Oxford

Singh, J. (1972) Great Ideas of Operations Research. New York: Dover

Softtech, Inc. (1981) Integrated Computer-Aided Manufacturing (ICAM) Architecture Part II: Function Modelling Manual, Wright-Paterson AFB, Ohio

Sorensen, R. (1995) A Comparison of Software Development Methodologies, Software Technology Support Center, <http://www.stsc.hill.af.mil/crosstalk/>

Spalazzi, L. (2001) A Survey on Case-Based Planning, Artificial Intelligence Review, Vol.16(1), pp.3-36

Stacey, M.K. and Eckert, C.M. (2003) Against Ambiguity, Journal of Computer Supported Cooperative Work (CSCW), Vol.12 (2), pp.153-183

Steward, D. (1981) The design structure system: a method for managing the design of complex systems. *IEEE Trans. on Engineering Management*, EM-28(3), pp.71-74.

Streufert, S. and Swezey, R. (1986) *Complexity, managers, and organizations*, Orlando; London: Academic Press.

Suchman, L.A. (1987) *Plans and situated actions: The problem of human-machine communications*. Cambridge, UK: Cambridge University Press.

Suh, N.P. (2001) *Axiomatic Design: Advances and Applications*. Oxford University Press

Sussman, G.J. (1975). *A Computer Model of Skill Acquisition*. American Elsevier, New York.

Wallace, K.M. and Blessing L.T.M. (1999) Visions for design research, in *Critical Enthusiasm: Contributions to Design Science*, Edited by Mortensen, N.H. and Sigurjónsson, J. Department of Control and Engineering Design, Danish Technical University, Denmark, pp.193-199

Waldrop, M.M. (1993) *Complexity: The Emerging Science at the Edge of Order and Chaos*. Simon and Schuster

Waterman, H., Tillen, D., Dickson, R. and De Koning, K. (2001) Action research: a systematic review and guidance for assessment. *Health Technology Assessment* 2001, Vol. 5: No 23.

Weld, D. (1994) An Introduction to Least Commitment Planning, *AI Magazine*, 15(4), pp.27-61

Whetten, D.A. (1989) What constitutes a theoretical contribution? *Academy of management review*, 1989, Vol.14(4). pp. 490-495

Whitney, D.E. (1993) Nippondenso Co. Ltd: A Case Study of Strategic Product Design, Research in Engineering Design, Vol. 5, pp.1-20.

Wilf, R.K. (2002) Algorithms and Complexity, ISBN: 1-56881-178-0, AK Peters limited

Wynn, D. and Clarkson, P.J. (2004) Models of designing pp.34-59 in Clarkson, P.J. and Eckert, C.M.(eds.), Design process improvement, Springer, 2004.

Wynn, D., Clarkson, P.J. and Eckert, C.M. (2005). A model-based approach to improve planning practice in collaborative aerospace design. ASME DTM 05, Long California, USA

Wynn, D., Eckert, C.M and Clarkson, P.J. (2006). Applied Signposting: A modelling framework to support design process improvement. ASME DTM 06. Philadelphia, Pennsylvania, US (accepted)

Yassine, A., Falkenburg, D. and Chelst, K. (1999) Engineering design management: An information structure approach, International Journal of Production Research, Vol.37

Yassine, A., Whitney, D. and Zambito, T. (2001) Assessment of Rework Probabilities for Design Structure Matrix (DSM) Simulation in Product Development Management, ASME DTM 2001, Pittsburgh, Pennsylvania, US

Yin, R.K. (1993) Applications of Case Study Research (Applied Social Research Methods). Sage Publications. Newbury Park, London, New Delhi

Zeigler, B.P. (1976) Theory of modelling and simulation, Wiley – Interscience, New York.