INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15-18, 2005

MODELLING AND SIMULATING ITERATIVE DEVELOPMENT PROCESSES

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

Keywords: process modelling, planning, Signposting.

1 Introduction

The development of a complex product can involve the co-ordination of hundreds or even thousands of individuals, each performing tasks which may span a range of disciplines, companies, and locations. These tasks form a shifting, iterative network of interrelated activities, which design managers strive to understand in order to deliver products to time, budget and quality constraints. Unfortunately, those planning methods used in industry are often ill-equipped to develop and maintain information about this type of dynamic process; as a result, managers must often make decisions based on a limited overview of the development process and its associated risks. Currently, such overview is often derived from many overlapping schedules and process representations which are developed individually by designers and design team leaders.

We are carrying out a programme of research to develop methods and tools to model and improve complex, iterative design processes. In this paper, we describe a pragmatic method which helps to develop a common understanding of process overview through the use of process modelling to support design planning at the operational level. The method was developed during an extended on-site case study carried out at a major UK aerospace manufacturer. Based on the ‘Signposting’ approach, a hierarchical model has been developed in which the design process is described in terms of tasks and their input/output parameters, together with resource requirements and uncertainty information. We describe how simulation techniques may be used to develop schedules from such process models, and to present them in a form useful to inform planning. The emphasis of the paper lies on the development of this pragmatic approach and the prototype software in which it is implemented.

1.1 Objectives

The research described in this paper was initiated during an extended on-site case study at a major UK aerospace manufacturer. The primary objective of the study was to develop a method to support more detailed task planning in the complex design processes found in the company. It was believed by the project management personnel that a poor level of granularity in design plans was contributing to difficulties in monitoring and controlling the progress of the project; in particular, it was felt that improved process visibility would allow schedule slippages to be identified and corrected before an avalanche of knock-on consequences could occur.

One cause of difficulties in planning design activities is the unpredictable nature of designing. Design processes are subject to influence from many factors, including: changes in the design team’s focus of attention, corresponding to engineering judgements which are made as the solution emerges; the unpredictability in lead times of subcontracted design work; and the frequent introduction of new design technology. However, complex product development is also subject to many technical constraints. These constraints stem from factors such as the need to utilize previous experience, to use specialist analysis tools, suppliers and to conform to product platforms. In one perspective of the design process, the combination and interaction of these constraints acts to form a ‘backdrop’ of possible process routes which, although continually evolving, may be considered relatively stable on the shorter timescale of the component design process studied in this work.

Early in the study described in this paper, it became clear that the poor granularity of design plans was caused in part by shortcomings in the standard Gantt chart planning package used by the company - the engineering teams’ detailed process knowledge could not be easily captured and used to inform planning activities. It was thus agreed with the company that the research would investigate: 1) the application of process modelling techniques to capture the stable backdrop of design tasks and their interactions in terms of information flow, iterations, and key programme milestones; and 2) the manner in which such a model might be used to support dynamic planning as the project unfolded.

1.2 Methods

The research described in this paper is primarily based on a case study during which eight months was spent on-site by the first author. The work also draws on experience gained during a number of previous studies in the aerospace and automotive sectors [2][3]. The new approach has been developed through an iterative process of critique and refinement, during which time the researcher attended meetings, conducted informal interviews and worked closely with the engineering and management personnel involved in component development, process modelling and design planning in the project. The unusual length and ethnographic elements of the case study allowed feedback from domain experts to strongly influence the direction of research; the issues addressed were thus thought to be of relevance and importance by both engineers and managers.

2 Overview of the approach

2.1 Process documentation in industry

During the study reported in this paper, a number of representations of the design process were collected as a precursor to developing the new planning method. A common form was the technical ‘process map’, almost exclusively confined to a single discipline and constructed using a standard office productivity tool. These documents captured design tasks and their interactions, often highlighting key software packages together with data types and other design descriptions. Many followed a ‘flowchart’ format, indicating information flow or a sequential progression of tasks and capturing limited possibilities for rework as cycles on the diagram. These documents were produced by members of the design teams, usually to represent technical process information relevant to a small number of other personnel. There were few standards of notation or format between these very specific descriptions, which were often constructed to highlight ambiguous or risky elements of the process. Most of the documents were considered to be out of date to some degree; during the study, they proved useful nonetheless to guide discussions and in reasoning about the nature of the design process.

The limited audience and specialist function of such descriptions, combined with constant time pressures, appeared to provide little incentive for co-ordination of documentation activities. A more integrated picture was provided by the company's intranet site, which provided a more prescriptive view of the development process. However, this information was thought to provide less useful guidance for technical design activities. One offered reason was the effort required to maintain and update such documentation in the face of evolving design technology in the company.

2.2 Design planning in industry

An empirical picture of planning in industry is provided by Eckert *et al* [4], who carried out research studies in a number of design companies. They describe how many documents were used in parallel to inform planning activities, and how these representations took on a variety of forms – including individuals' activity checklists, the ubiquitous Gantt charts, process maps, and even bills of materials. The information content of these documents was found to exhibit high overlap and a low degree of coherence. Eckert *et al* concluded that global consistency in planning is achieved through an ongoing process in which many individuals reason about and maintain overlapping sets of representations, and that the corresponding lack of overview can lead to avoidable mistakes and inefficiencies.

2.3 Approach

A number of methods have been proposed to model the design process and/or improve planning practice in industry. Although these methods are often received with enthusiasm during research case studies, relatively little impact has yet been made on wider industrial practice. In this study, we aim to address some shortcomings of previous research by developing a method and software tool which are attractive to the designers and design teams responsible for design and for process documentation, in addition to those personnel concerned with planning. In following sections, we describe how a graphical process modelling framework and supporting software tool has been developed to support the integration of existing process documentation activities with design planning.

The availability of effective, intuitive software tools is important to develop process models which contain a realistic level of detail. In this study, process modelling software was continuously prototyped and evaluated in conjunction with the modelling method and framework; this iterative approach helped to ensure the viability of the underlying concepts as well as the effectiveness of the tools themselves.

2.4 Overview

Our approach consists of the following activities:

1. Detailed knowledge about the design process is elicited and modelled using an intuitive flowchart format, resulting in a library of 'building block' processes. The format allows the explicit specification of possible rework or design iteration as static, cyclic dependencies between tasks.

2. The ‘building block’ processes are manually arranged to describe the anticipated plan of work, while satisfying any parametric dependencies which constrain routes through the design process. Iteration is implicitly described by the re-use of ‘building blocks’ throughout the plan.

3. Task durations are estimated using triangular probability density functions, and the explicit rework cycles are parameterized in terms of numbers of planned iterations or estimated likelihoods of failure.

4. A *milestone constraint* is specified, describing the date by which the work must be completed.

5. The parameterized plan is analyzed to automatically generate a representative schedule that meets the specified information flow, task duration and resource constraints. The schedule has an associated *schedule risk*, corresponding to the likelihood that the process will run beyond the specified milestone.

To illustrate this method, figure 1 depicts a section of a planning network and two representative schedules in Gantt chart format. The topmost Gantt chart shows a medium risk schedule generated from the planning network on the left; based on estimated task durations and a likelihood of rework on the evaluation task, up to two iterations may be carried out and still meet the specified milestone. In a common scenario, a longer period of time has been dedicated to the first task than originally anticipated; the bottom Gantt chart depicts a different schedule which may be generated from the same planning network.

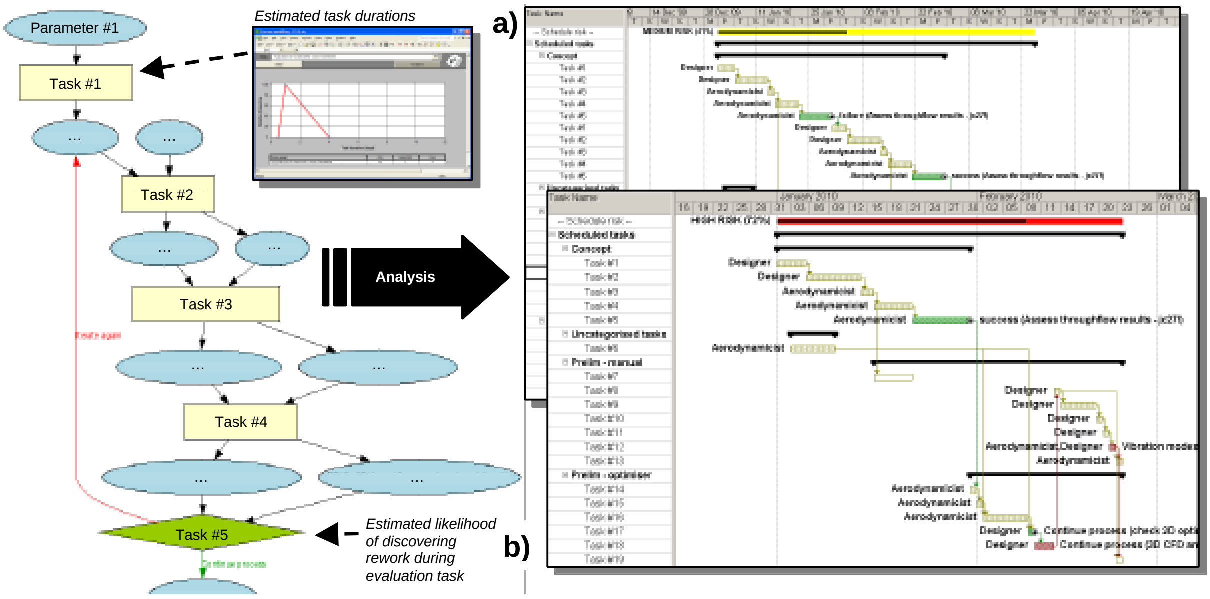


Figure 1. A fragment of a cyclic plan and two representative schedules, showing: a) two scheduled iterations and ‘medium’ schedule risk; and b) one scheduled iteration and correspondingly ‘high’ risk

The updated schedule reveals that only one pass at the tasks on the left may now be made in order to meet the same milestone. The less ideal situation is reflected in a higher value of estimated schedule risk. The team may accept this increased risk; or may choose to re-evaluate the plan of work to reduce it. This may involve planning to spend less time on future tasks than previously anticipated; reconfiguring the arrangement of processes which form the plan to incorporate fewer iterations; or even removing certain tasks altogether.

This example illustrates how, through an ongoing process of iterative refinement, steps 1-5 may be used to develop and maintain a description of the plan of work that, subject to appropriate characterization and modelling of process behavior, may be made more detailed and more robust to uncertainty in task durations and outcomes than the Gantt-based plans in common use. In the case study, further benefits of the method were thought by company personnel to include: 1) the collaborative, interdisciplinary process of modelling and plan development, which was thought to '*get people talking*'; and 2) the development of prototype tools for capturing and manipulating detailed process information. In the latter case, the prototype tools provide the capability to develop much larger and more detailed process models than are feasible using the standard office package, and to manipulate and visualise these models in an appropriate fashion.

3 Implementation of the approach

3.1 Existing process modeling frameworks

A number of frameworks have been proposed for modelling the design process [5]. Some use formal graphical notation - for example, Kusiak *et al* used IDEF0 to model design activities, also capturing the flows of information and resource through the process [6]. Many others are based on the DSM [7][8]. These may be used to capture the effects of task ordering on process effectiveness [9]; extensions capture uncertainty in task durations and outcomes [10]. The model proposed by Krishnan et al [11] may be used to describe situations in which the boundaries of design tasks overlap. These examples all describe the design process in terms of direct relationships between tasks. Other schemes capture process behaviour by explicitly capturing information flow between tasks. For example, McMahon and Xianyi [12] used Petri nets to automate repetitive crankshaft design by capturing the flows of information between computer tasks. Dynamic frameworks such as Signposting [2] and the Adaptive Test Process [13] describe design tasks in terms of input and output parameters, where the term 'parameter' may be used to refer to a description of any aspect of the product or process which changes over time; this includes data files and design reports as well as numerical parameterizations.

The Signposting framework characterizes designing as the identification and iterative refinement of parameters. Design processes are represented as a set of *parameters* and *tasks*, each of which is defined in terms of one input *state* and one or more output states. An input state describes the parameters used by the task, together with a numerical description of the minimum level of *confidence* in each which is deemed appropriate to starting the task. Similarly, output states describe the parameters which are produced when the task is completed, together with the level of confidence which the task lends to each parameter. At any time, the state of a process may be represented by a vector describing the level of confidence in each parameter. The tasks which are possible to begin at that time may be determined by comparing the input state of each task against the current level of confidence in each parameter; the task may begin if each parameter in the task's input state is available to at least the specified level of confidence, and if at least one transition would result in an increase in the Euclidean length of the state vector. Signposting models thus describe a dynamic process of task selection, capturing aspects of the inherent, solution-oriented uncertainty of the design process. As a consequence of specifying processes in terms of knowledge about individual tasks, Signposting models typically capture a wide range of possible process routes. This allows exploration of many alternative configurations and 'what-if?' scenarios; a key advantage of Signposting which has previously been exploited in process optimization applications [14]. The Signposting framework is especially appropriate in the modelling of adaptive or variant design processes where the majority of tasks and parameters are well delimited and may be identified in advance. This is the case with the component design process studied in this work, which may be described to a large extent in terms of the use of design and analysis tools.

3.2 Development of the Applied Signposting Model

The Signposting model was chosen as the basis of this research due to its dynamic approach, detailed task definitions and flexibility. However, extensions have been made due to two observations regarding task selection in collaborative, parameter-driven design processes: firstly, although a very large number of parameters are used throughout the course of the process, only a relatively small number drive the dynamic selection of tasks; and secondly, although the design process must conform to the ‘hard’ constraints of data requirements of design and analysis tools, in reality only a small set of these possible routes are likely to occur in practice. The extensions described below are designed to support the modelling and later manipulation of this anticipated subset of routes.

As with the Signposting approach described above, the Applied Signposting Model (ASM) is constructed from *tasks* and *parameters*. However, the ASM provides three types of task: *Simple tasks*, which link one input state to one output state and may be used to represent activities whose outcome is assumed to have no direct effect on the choice of following tasks; *Iteration construct* tasks which contain one input state and exactly two output states, and are used to represent evaluation-type tasks which may result in the discovery of additional rework; and *Signposting tasks*, which contain one input state and many output states, selected stochastically according to estimated probabilities of task outcome. Signposting tasks may be used to encode a very wide range of possible process scenarios, as described in detail by O’Donovan [20]. However, in this study it was found that many design activities could be adequately represented using a much simpler definition; the two new task types were introduced to improve ease of modelling in these cases.

In addition to input and output parameters, any requirements for resources which are drawn from common pools for execution of the task may be specified, together with best case, most likely, and worst case estimates of task duration. This treatment of uncertainty was chosen due to its familiarity to the engineers and managers involved in the study. The task definitions are summarised graphically in figure 2; the use of this colour and shape coding scheme throughout the modelling tools allows the easy identification of task behaviour from any view of the process.

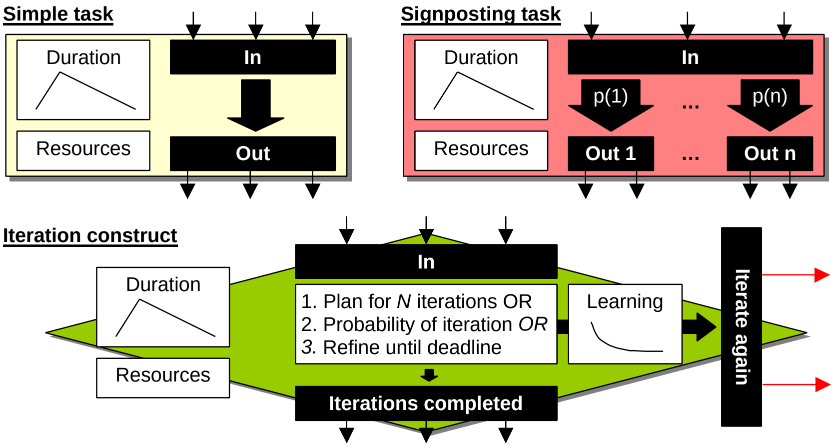


Figure 2. Applied Signposting Model (ASM) task definitions

As with previous generations of the Signposting model, tasks are defined within the context of parameters. However, in the ASM extension both tasks and parameters are defined within the context of a *process*, defined as a set of tasks and parameters together with a single input and single output state drawn from the parameters within the process. This scheme allows the hierarchical definition of models in which tasks are dynamically selected according to the confidence state of parameters in their parent process, rather than the global list of parameters. Design processes are thus described in terms of: 1) *knowledge* about individual tasks and their input/output characteristics; and 2) *assumptions* regarding the limited scope of a task's effect upon other tasks in the model. In addition to enabling effective navigation of process visualisations and supporting the structuring of knowledge elicitation activities, this hierarchical approach provides a very flexible means to capture the iterative behaviour of the design process.

A key component of the ASM is the graphical notation depicted in figure 3. In this notation, ellipses are used to denote each possible combination of parameter and confidence within a process. This allows an ASM model to be constructed either from knowledge of individual tasks or, as is typically the case in the method described here, as a process flowchart capturing a procedural overview. In either case, the model may be presented and manipulated as an information flow diagram. With appropriate software support, this visualization allows the description of processes in an intuitive format which conforms to the formal syntax of the ASM. However, in comparison to a flowchart drawn using a standard office package, this formality allows an ASM model to be 'executed' using the simple algorithm described in the forthcoming *analysis* section.

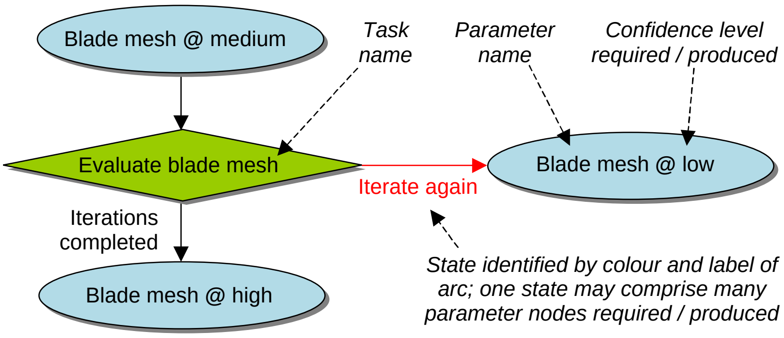


Figure 3. Applied Signposting Model (ASM) graphical notation

3.3 The modelling activity

Process knowledge may be elicited from many sources, including human experts, explicit procedures and other documentation. Each source typically represents a limited perspective of the process; as a result, no single acquisition technique may be uniformly applied. This 'bottleneck of knowledge elicitation' is well known [15]. In this study, the researcher acted as facilitator throughout the modelling process, which was driven by the design team leader responsible for timely delivery of the component. This arrangement was influenced by the requirement to develop the new method and prototype software concurrently with the model itself. Expert knowledge regarding the detail and required scope of the process model was elicited through a combination of informal discussions with technical and management personnel, study of existing documentation and a small number of group workshops.

This modelling activity resulted in the development of a library of 'building block' processes, a subset of which may be seen in figure 5. These building blocks were then used to compose the process model; since several areas of activity were considered very similar at the chosen level of detail, a lattice structure was used in place of a simple hierarchy to avoid duplication of modelling effort (figure 4). Uncertainty inherent to the process was modelled at three distinct levels: At the lowest level, durations of certain design and analysis tasks were described using triangular probability density functions; iteration constructs within process blocks were used to represent both possible rework cycles arising from inadequacies discovered during data integration and validation activities and planned cycles of iterative refinement; and at the highest level the dynamic task selection features of the ASM were used to model the possibilities for arrangement of the top-level sub-processes to form a plan. The latter case was used to capture the effect of the team's in-situ design decisions in determining the course of the process.

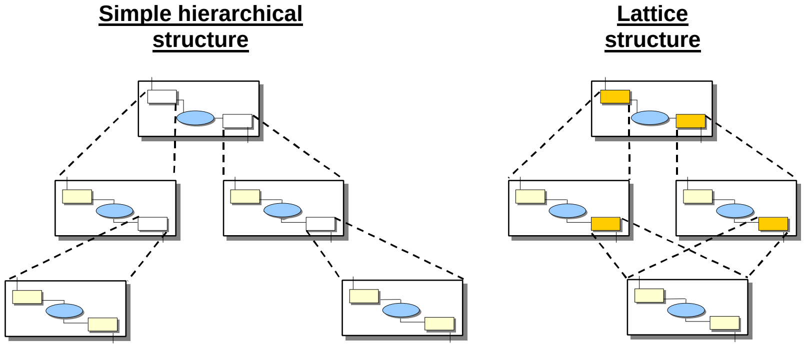


Figure 4. Use of 'building block' processes to compose a lattice structured process model

Previous to this study, no integrated picture of low-level design activities existed for the component; the knowledge elicitation and modelling process was thus a highly iterative process of critique and gradual refinement which continued over a period of several months. This type of detailed modelling places different requirements on tools and methods than less effort-intensive approaches; in particular, although a structured approach to modelling and knowledge elicitation was initially important, the development of an appropriate, intuitive, graphical framework and effective support tools were key to the company's ongoing support of the approach.

3.4 Development of prototype modelling software

During this study, prototype software was developed to provide three interactive visualizations of the ASM model elements: a view showing the input and output states of individual tasks; a matrix view, showing the tasks in a process and the parameters they require and produce; and an interactive network view illustrating information flow in a process (Figure 5).

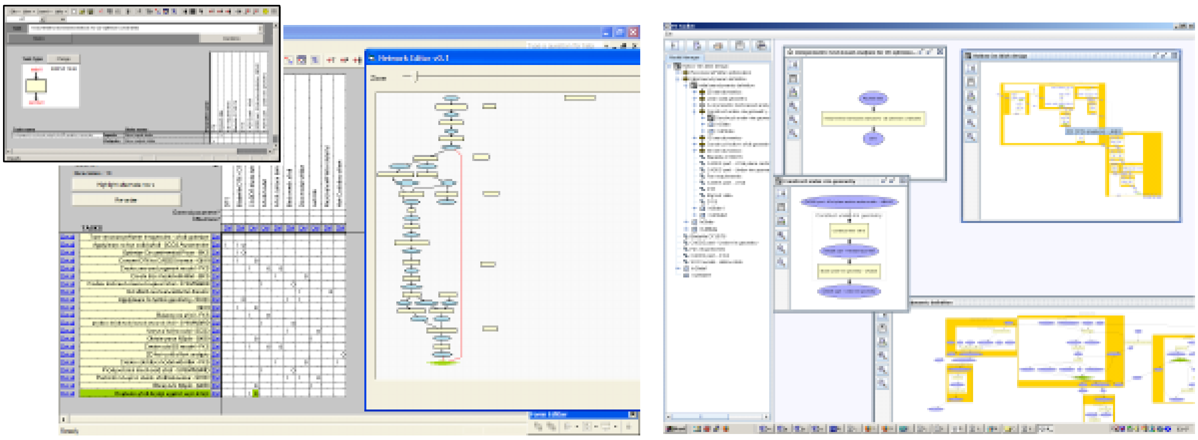


Figure 5. (left) Initial prototype software showing task state, task/parameter matrix and network visualizations (in left to right order); and (right) re-implementation for further evaluation and application.

Such visualizations of process models can serve a number of purposes, including: aids to verbal or written communication about the process; placeholders for the modeler’s knowledge; and, if sufficiently interactive, as a tool for knowledge elicitation. Just as a model should be developed to provide a useful abstraction of reality, so a visualisation should be designed to provide a useful perspective of the underlying model. During the course of the study, the form of visualization chosen for use during knowledge elicitation and model construction was found to form a strong influence upon the resulting process description. For example, the use of a matrix format during group elicitation sessions has been reported useful in providing structure to the exercise [16], a finding repeated in this study; however, when asked to describe process behavior individually or in very small groups, engineers always sketched diagrams using a flowchart format. Furthermore, procedural data elicited or modelled using the task/parameter matrix or task state views resulted in a very strongly connected process, typically capturing a large number of possible routes. However, when presented with a network visualization, the same users found ways to describe their process knowledge in more linear fashion which resulted in a visually appealing diagram. In this case, it may be concluded that network views are appropriate to the capture of procedural information whereas matrix-based approaches are effective in drawing out knowledge of interdependencies. However, even apparently minor variations can have a strong influence upon the resulting models – in one example, an automatic, compact graph layout algorithm with effective edge routing has been found to encourage detailed task and parameter description more than a similar algorithm giving less compact and less visually pleasing results.

Network layouts are currently generated automatically using the *dot* algorithm distributed as part of the GraphViz package [17]. Although this approach has a number of shortcomings, modelling tools based on fully automated layout systems can be particularly effective when manipulating larger models, relieving the user of a substantial manipulation burden. In particular, this type of system appears to dramatically reduce the barrier to modelling via incremental manipulation of process data.

It was also observed that the common use of many, overlapping process representations was not purely due to a deficit of suitable modelling tools; whether by accident or design, each of these visualizations served a specific purpose. For example, a process map developed by aerodynamicists provided a highly compressed description of non-aerodynamics tasks; later, a full printout of the process network developed in the study was found to encapsulate the information content but obscure the message of the original slide. This observation led to the introduction of configurable *classification schemes* for tasks and parameters and the development of code to automatically generate network visualizations. This allows the interactive modelling display to be manipulated in sophisticated ways, by specifying node shape, color and border according to task type and using Boolean operations to filter the resulting network. To illustrate, a scheme entitled ‘discipline’ was created and used to classify tasks into the categories of ‘aerodynamics’, ‘stress’ and ‘design’. Likewise, the ‘stage’ scheme consisted of the ‘prelim’, ‘2D’ and ‘3D’ categories. Using the software, the user may easily obtain a customized visualization of the process as a network of tasks in the ‘prelim’ stage, of which ‘aerodynamics’ tasks are highlighted in a certain color.

3.5 Scheduling

A number of authors have described methods that aim to explore or improve design practice through the analysis and optimization of process models. These include the use of simulation to compute the probability distributions of quantities such as risk, cost and lead time [18]; the calculation of stochastic measures of task criticality [19]; and the development of probabilistic schedules [14]. In following paragraphs we describe a simple analysis technique which may be used to resolve an ASM model into a schedule which satisfies a specified set of task duration, resource and milestone constraints. This technique has three components: firstly, an algorithm for executing a plan constructed using the ASM framework; secondly, a simple simulation algorithm; and thirdly, a method to select an appropriate representative run from the resulting population of possible processes. Unlike the ASM model from which it is generated, the single representative process is acyclic and easily rendered in familiar form as a Gantt chart highlighting task duration milestones, dependencies and resource requirements.

The primary requirement of the execution algorithm is to operate effectively upon an ASM process described using the intuitive graphical notation described in previous sections. In order that ASM flowchart diagrams may be executed in the expected manner, as determined by their intuitive interpretation, an alternative algorithm for the selection of available tasks in ASM processes has been developed based on the Petri net approach to modelling information flow used by McMahon [12] and the ‘impulse’ method described by Andersson *et al* [20]. In our approach confidence values are treated as non-numeric, allowing their definition in terms of an appropriate textual description. Describing the algorithm in terms of the graphical notation of the ASM, each task in an ASM process corresponds to a Petri net transition, and each input arc to the task is associated with a *place*. In contrast to standard Petri net places, ASM places may take one of three states: *inactive; active;* and *tokenized*. Upon initialization of the simulation algorithm, all places are made inactive except those that are not connected to a task via a parameter-confidence node (or, when simulating a hierarchical process, a chain of parameter-confidence nodes); these are made tokenized on initialisation. At any point in time, all tasks are categorized as either: *not possible*, if one or more input places are inactive; *possible*, if all input places are active; or *recommended*, if at least one input place is tokenized and no input places are inactive. Only recommended tasks are available to begin. Upon completion of the task, all input places are made active and all output places are made tokenized. To illustrate this algorithm, an example step is shown in figure 6.

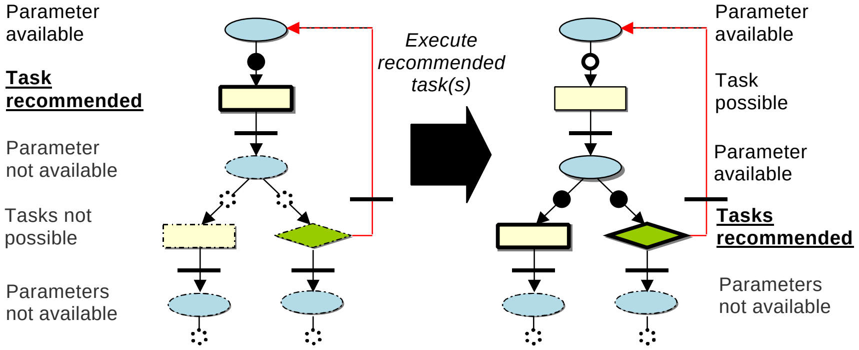
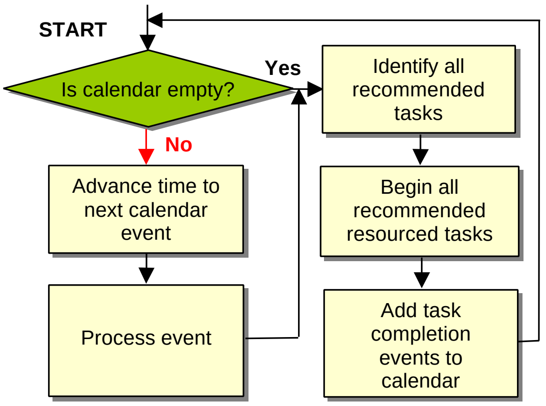


Figure 6. ASM model showing tri-state places (circles: solid fill=tokenised; solid border=active; dashed border=inactive) and transitions (horizontal bars)

The execution method described above allows the simulation of an ASM plan using a straightforward discrete event algorithm summarised in figure 7. In addition to constraints on the ordering of tasks, the simulation accounts for resource constraints and the uncertainties specified in the durations of tasks and in likelihoods of rework. Execution of the simulation allows computation of a population of possible process routes which, in this work, are used in simple fashion to develop a schedule of design tasks. This type of simulation may also be used for more advanced analysis of ASM models; such techniques are described in detail by O’Donovan [1].



Recommended tasks:

All input places *available*
AND
One or more inputs *tokenised.*

Begin task:

Set all input places to *available*
AND
Remove resources from pools.

Complete task:

Set all output places *available*
AND
Set all output places *tokenised*
AND
Return resources to pools.

Figure 7. Simplified simulation algorithm for an ASM model

O’Donovan [14] describes how scheduling information may be extracted from a population of possible processes and displayed using probabilistic timing diagrams, which provide a view of the likelihood that each task will be in execution at any given time. While this method can provide a good description of process properties, for the purposes of this work it is necessary to provide a schedule which also provides a prescriptive perspective; that is, to answer the question: *In order to meet a certain milestone for completion of the process, by when must I complete each constituent task?* To achieve this a set of candidate processes is selected from the population resulting from simulation, and the most representative is extracted from this set. Selecting a single process run for visualisation as a Gantt chart allows the depiction of detailed task information which may be directly related to the relationships between tasks captured in the original, cyclic plan. This graphical relationship, clearly illustrated in figure 1, allows the process model to be easily manipulated in order to achieve desired changes to the resulting schedules.

The naïve heuristic currently used to obtain a representative set of processes based on a given milestone constraint is described in figure 8. The 'most representative' member of this set is currently chosen by calculating the mean number of repetitions of each task across all processes in the set, and selecting the single process which is closest to matching this mean. Finally, based on figure 8, the schedule risk displayed in figure 1 is equal to the number of failed processes divided by the total number of simulated process runs, where a failed process is defined as a simulation run which exceeds the specified milestone date.

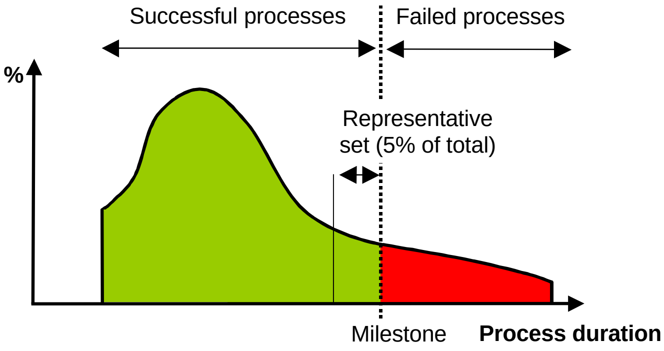


Figure 8. Extracting a representative process from the population of possibilities

4 Evaluation

It is difficult to define concrete success criteria for this research, due to the long timescales of the project (over 10 years from concept to full production) and the many influences on the effectiveness of design planning. However, at the conclusion of the study, the method had allowed the company to develop a model of design activities which exhibited significantly greater detail than was previously possible, from which schedules were automatically generated. This preliminary application thus suggests that the modelling approach and software concept may provide a useful capability to the company - at the request of both the programme manager sponsoring the planning research and the design team leader responsible for delivering the component, the prototype modelling and analysis tool developed as part of the study has been re-implemented to allow further evaluation, development and application (figure 5, right). At the time of writing, this tool has been in use in the company for one month; more detailed evaluation is planned based on user feedback. Ultimately, however, the most appropriate success criterion will lie in the company's continued support of the research and use of resulting tools.

5 Conclusion

This paper has described the development of a pragmatic method to support planning practice in industry through more detailed task planning in the complex design process. Preliminary evaluation of the method has been positive, conducted via application and feedback during the case study. At the request of company personnel directly involved with the research, the process modelling and analysis tool developed as part of the research has been re-implemented and is currently undergoing further application and evaluation.

Software prototyping and user interface development has formed an ongoing and non-trivial challenge which has had significant impact on the effectiveness of this research. In addition the development of the software has highlighted a number of important areas for further research. Notable examples which we hope to investigate include the development of computationally efficient, incremental methods for the layout of process networks (maintaining similar spatial relationships between nodes and edges to assist the user in recognising features) and for run-time feedback to ensure that all process models constructed using the software are suitable for automatic analysis using simulation techniques (for example, that such models do not contain impossible tasks or interdependency ‘deadlocks’). Other areas for future work include the application of process optimisation methods to improve the naïve method used for selecting a representative process for scheduling, and the development of more effective visualisation techniques for planning with a focus on describing parallel task streams.

References

[1] O’Donovan, B. D. O., “Modelling and Simulating Design Processes”, PhD thesis, University of Cambridge Department of Engineering, 2004.

[2] Clarkson, P. J. and Hamilton, J. R., “Signposting: a parameter-driven task-based model of the design process”, Research in Engineering Design, 12(1), 2000, pp. 18-38.

[3] Eckert, C. M., Clarkson, P. J. and Zanker, W., “Change and customisation in complex engineering domains”, Research in Engineering Design, 15(1), 2004, pp. 1-21.

[4] Eckert, C. M. and Clarkson, P. J., “The Reality of Design Process Planning”, ICED03 Stockholm, 2003.

[5] Browning, T. R. and Ramasesh, R. V., “Modelling the Product Development Process: A Review” TCU M.J. Neeley School of Business Working Paper, 2004.

[6] Kusiak, A., Larson, N. T. and Wang, J., “Reengineering of Design and Manufacturing Processes”, Computers and Industrial Engineering, 26, 1994, pp. 521-536.

[7] Austin, S., Baldwin, A., Li, B., and Waskett, P., "Analytical Design Planning Technique: a model of the detailed building design process." Design Studies, 20(3), 1999, pp. 279-296.

[8] Steward, D. V., “The design structure matrix: A method for managing the design of complex systems,” IEEE Transactions on Engineering Management. EM-28(3), 1981, pp. 71-74.

[9] Eppinger, S., Whitney, D., Smith, R. and Gebala, D., “A Model-Based Method for Organizing Tasks in Product Development”, Research in Engineering Design, 6, 1994, pp. 1-13.

[10] Carrascosa, M., Eppinger, S. D. and Whitney, D., “Using the design structure matrix to estimate product development time,” Proceedings of ASME Design Engineering Technical Conferences, Atlanta, GA, USA, 1998.

[11] Krishnan, A., Eppinger, S. and Whitney, D., “A Model-Based Framework to Overlap Product Development Activities,” Management Science, 43, 1997, pp. 437-451.

[12] McMahon, C. A. and Xianyi, M., “A Network Approach to Parametric Design Integration”, Research in Engineering Design (1996) 8, 1996, pp. 14-32.

[13] Levardy, V., Hoppe, M., Browning, T. R., “Adaptive Test Process – An Integrated Modelling Approach For Test And Design Activities In The Product Development Process”, Proceedings of ASME Design Engineering Technical Conferences, Salt Lake City, Utah, USA, 2004.

[14] O’Donovan, B. D. O., Eckert, C. M. E. and Clarkson, P. J., “Simulating Design Processes to assist design process planning” Proceedings of ASME Design Engineering Technical Conferences, Salt Lake City, Utah, USA, 2004.

[15] Buchanan, B. G. et al., “Constructing an Expert System,” in “Building Expert Systems”, Hayes-Roth, F. et al, Addison Wesley, 1983.

[16] Jarratt, T.A.W., “A model-based approach to support the management of engineering change,” Ph.D. Thesis, University of Cambridge, Department of Engineering, 2004.

[17] GraphViz, http://www.graphviz.org/

[18] Browning, T. R. and Eppinger, S. D., “Modelling Impacts of Process Architecture on Cost and Schedule Risk in Product Development,” IEEE Transactions on Engineering Management, 49(4), 2002, pp. 428-442.

[19] Cho, S. and Eppinger, S. D., “Product Development Process Modelling Using Advanced Simulation” Proceedings of ASME Design Engineering Technical Conferences, Pittsburgh, Pennsylvania, USA, 2001.

[20] Andersson, J., Pohl, J. and Eppinger, S. D., “A Design Process Modelling Approach Incorporating Nonlinear Elements”, Proceedings of the ASME Design Theory and Methodology Conference (DETC 1998), Atlanta, 1998.

David Wynn
Engineering Design Centre
University of Cambridge Engineering Department
Trumpington Street
Cambridge
CB1 1PZ.
United Kingdom.
Phone: +44 1223 332828
Fax: +44 1223 766963
E-mail: dcw24@cam.ac.uk