A MODEL-BASED APPROACH TO IMPROVE PLANNING PRACTICE IN COLLABORATIVE AEROSPACE DESIGN

David Wynn  
Engineering Design Centre  
Department of Engineering,  
University of Cambridge,  
Trumpington Street, Cambridge.  
CB2 1PZ. U.K.  
dcw24@cam.ac.uk

P. John Clarkson  
Engineering Design Centre  
Department of Engineering,  
University of Cambridge,  
Trumpington Street, Cambridge.  
CB2 1PZ. U.K.  
pjc10@cam.ac.uk

Claudia Eckert  
Engineering Design Centre  
Department of Engineering,  
University of Cambridge,  
Trumpington Street, Cambridge.  
CB2 1PZ. U.K.  
cme26@cam.ac.uk

ABSTRACT

The research reported in this paper complements previous work in the modeling and analysis of product development processes to reduce process risk and lead time. Such methods have found a receptive audience in industry, as evidenced by many academic case studies and the initiation of similar process improvement initiatives by the companies themselves. Given the ongoing ‘pull’ from a number of companies, we are interested to understand why such model-based methods have not yet impacted upon wider operational planning in industry.

In this paper, based on an extended research study at a large UK aerospace company, we describe the application of the Signposting process modeling approach to the problem of developing a pragmatic method to support planning practice. The research described in the paper focuses on the elicitation, representation and manipulation of process information in a form which can inform planning and scheduling; during the study, addressing these issues was key to gaining support of the approach within the company.

INTRODUCTION

The research described in this paper is based on a long-term case study carried out in a large UK aerospace company. The study was initiated after the company carried out a ‘lessons learnt’ exercise following completion of an early phase of a major new development project. The exercise had highlighted a recurring issue thought to be of particular importance; the design-make plans for major components would detail manufacturing processes in great depth, down to the level of transferring components between job shops, whereas the design phase was represented at far poorer resolution – design tasks were planned on the scale of months instead of days.

The programme manager sponsoring the research hoped to unpack these ‘black box’ design plans – it was envisaged that this would allow a detailed checklist of design activities and deadlines to be constructed and maintained as the project unfolded. This would lead to more effective progress monitoring, allowing schedule slippages to be identified and remedial action to be taken before an avalanche of consequences was triggered.

After conducting some preliminary observations and informal interviews with the personnel concerned it was concluded that available planning and modeling techniques suffered substantial shortcomings in describing the behaviour of the complex design processes found in the company. In particular, the acyclic networks upon which the standard resource planning and Gantt-based scheduling packages are based are inappropriate to represent the iterative behavior of typical design processes. In such cases, uncertainty in the characterization, ordering and duration of design tasks is driven by the inherently opportunistic and responsive nature of designing. In practice, these uncertainties prevent the development and maintenance of a detailed, ordered list of design tasks and their expected durations.

A new type of plan was required, and, just as importantly, a new way of approaching the planning activity – a more structured method which would be effective under the conditions of iteration and uncertainty about tasks and their ordering in the design process. This paper describes the development and initial application of such a method.

OBJECTIVES

The primary objective of the study was to develop a method to support more detailed task planning in the complex design process. This capability would improve the company’s
ability to manage and, ultimately, control their processes. To achieve these objectives, a process model would be constructed to capture information flow, resource usages, design iterations and key programme milestones. Simulation techniques would be applied to this model to determine ‘best case’, ‘most likely’ and ‘worst case’ schedules, which would show task durations measured in days and replace the current generation of highly ambiguous, coarse-grained Gantt charts. Furthermore, it was envisaged that by automating the generation of these schedules from the ‘master model’, planning would become less costly and more robust to uncertainty about the process. These objectives would be achieved through the application and, where necessary, modification of existing methods.

This paper describes the development of a new approach which meets these objectives. We have chosen to tell the story from an industrial perspective in order to illustrate the development of our integrated, requirements-driven approach in response to practical challenges encountered during the research.

RESEARCH METHODS

The case study described above has included eight months spent on-site by the first author. The research also draws on experience gained during a number of previous studies in the aerospace and automotive sectors [1, 2]. The study was split into two phases: firstly, two months were spent identifying and clarifying the research objectives; and secondly, six months were spent developing the new method through an iterative process of design, critique and refinement. Throughout this period the researcher attended meetings, conducted informal interviews and worked closely with the engineering and management personnel responsible for process modeling and design planning in the project. The unusual length and ethnographic elements of the case study allowed feedback from users to strongly influence the direction of research; the issues addressed were thought to be of relevance and importance by both engineers and managers.

BACKGROUND

The case study described in this paper focuses on the first stage fan blisk for a new military turbofan engine; a highly complex, integrative product. The fan blisk is a major subsystem that must meet a number of technical requirements. For example, it must exhibit certain aerodynamic properties as a part of the compressor module; it must resist the steady stresses caused by rotation and the periodic vibrations caused by interaction with other engine components. As the foremost rotative component, it must be especially resilient to impact from ingested matter. The blisk must also satisfy various materials and manufacturing constraints. From the earliest stages of design, issues of testing, manufacturability, maintainability and disposal must be addressed to ensure economic viability over the predicted lifecycle of the product.

The blisk design process

In the context of this paper, we are concerned with understanding the process that is followed during the design of a component, rather than with existing prescriptions or descriptions of practice.

Just as the form of the component is constrained by the need to satisfy many design objectives, so its design process is driven by the available design tools, facilities and experience. For example, understanding the relationship between the form of the blisk and any design objective requires a high degree of specialized knowledge. Consequently most aerospace engineers are trained in specialist disciplines and are expert in the use of particular tools to solve certain sub-problems. As a result, many possess only limited knowledge of the product outside their area of specialization. It may not be expected or necessary, for example, for an aerodynamicist to have a detailed understanding of the effect of a blade’s shape on its resistance to foreign-body impact. Furthermore, most complex products are platform based, have lifecycles measured in decades and are developed in large part by modification from existing designs. In this environment, radical designs carry a high risk and most designers only participate in a few truly novel projects during an entire career. The design process must also meet a number of business-oriented objectives. For example, contractual agreement may require the testing of prototypes on certain dates, requiring the release of intermediate designs in good time for manufacture. Many other limitations including the availability of human, computational and manufacturing resources also act to constrain the process.

Set against these stabilizing factors are a number of drivers of process uncertainty. For example, a key challenge in designing high performance products lies in combining specialist analyses and insights to find an appropriate trade-off between objectives. This is achieved through the iterative exploration of the design space and the convergent refinement of a diverse set of design descriptions. The direction in which the process unfolds is difficult to predict, as important design decisions are informed by an evolving understanding of the relationship between objectives and the form of the solution. This solution-oriented perspective is often considered key to understanding the nature of designing, regardless of context, discipline or relative complexity of the product [3]. Further, the design process is subject to the influence of many ‘external’ factors, such as the lead time of subcontracted design work. On a longer timescale, even the relatively stable aspects of the design process evolve as companies pursue advances in design technology and respond to influences such as changes in validation requirements and product/service economics [4]. The combination of these internal and external drivers of uncertainty can cause great difficulty in planning and managing large design projects. In summary, although complex design projects are constrained by a large number of factors, every project also contains elements of novelty in both product and process; this is a key driver of planning difficulties.

The members of a project team must manage many conflicting concerns when deciding how to meet their objectives. In reality, these management and design decisions are not centralized; as in any large organization, social negotiation and internal politics play a significant role in determining process behaviour [5]. Whereas design engineering often requires a detailed understanding of specialist methods and tools, design management thus requires overview of many issues which span the domains of product, process and personnel. In this paper, we are concerned with improving
Designing any complex product or component is a highly complex process which encompasses a wide range of skills, disciplines and constraints. The activities of design planning and modeling exhibit substantial complexity in their own right; complexity which builds upon that of the processes they aim to describe and control [7]. All the companies we work with expend significant effort in documenting their planning knowledge. Often taking the form of Gantt charts and process maps, such documentation provides a limited perspective of a company’s knowledge about their processes; knowledge which is distributed amongst the individuals who use and maintain it. Such knowledge is critical to effective planning, but is often subjective, partly tacit and thus difficult to elicit and represent. Due in part to the lack of coherent process documentation it can be difficult for design teams to reflect upon the activities which must be carried out; in such situations successful co-ordination of the design activities must often rely upon informal communication and the assumption of shared knowledge.

The study described in this paper has focused on a single engine component. Nevertheless, its design is a team activity which requires a number of highly specialized, geographically distributed members of the design IPT to trade technical improvements against the need to integrate their work to a tight schedule. A structured approach to support process documentation and planning would help the team in achieving this goal.

**AN APPROACH TO IMPROVE PLANNING PRACTICE**

The research described in the following sections aims to improve design planning with a pragmatic approach that supports the development of a detailed perspective of the relationship between technical design activities and the plan of work. In particular, the approach aims to integrate disparate process mapping activities within a formal modeling framework and use this information to inform planning practice. The framework is flexible enough to describe information previously captured in a diverse range of flowchart diagrams. To support the application of this approach, prototype software tools have been developed to enable models based on the framework to be constructed and manipulated.

The approach consists of the following activities:
1) Detailed knowledge about the design process is elicited and modeled using an intuitive flowchart format, resulting in a library of ‘building block’ processes. Possible rework or design iteration may be specified explicitly as 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’ in 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, resourcing and task duration constraints. The schedule has an associated schedule risk, corresponding to the likelihood that the process will run beyond the specified milestone.

6) Steps 1 to 5 may be repeated in a process of iterative refinement, until the plan is clarified and the perceived trade-off between the planned activities, their durations and iterations and the estimated level of schedule risk is deemed acceptable.

To illustrate this method, figure 1 depicts a section of a planning network and two representative schedules in Gantt chart format. The figure is drawn from data elicited during the study; however, to protect confidentiality the scheduling and risk information does not reflect the results of actual analysis. 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. The second Gantt chart depicts the scenario in which some time has passed, but a longer period of time has been dedicated to the first task than originally anticipated; a different schedule has been generated from the same planning network. The updated schedule reveals that only one pass may now be made in order to meet the same milestone. The less ideal situation is reflected in a high 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 modeling of process behavior, is more detailed and more robust to uncertainty than Gantt-based plans. In addition to structuring the planning activity, additional benefits of the method were thought by company personnel to include: 1) the collaborative, interdisciplinary process of modeling and plan development, which was thought to ‘get people talking’; and 2) the prototype software tools developed during the study, which enabled the capture and integration of technical process information on a larger scale than was feasible using existing software. While the research was initially targeted at programme management personnel,
these secondary benefits were critical in generating wider interest and support for the approach.

IMPLEMENTING THE PRAGMATIC METHOD

It is illuminating that the most successful techniques in industry embody simple, focused concepts which capture a very small number of factors and, as a result, are flexible enough to be applied to situations outside the original intent. Two classic examples are the widespread adoption of PERT-style process modeling and the popularity of Six Sigma methods. Conversely, we have studied a number of large process models resulting from very broad process modeling initiatives carried out by industry; these models can be so ambiguous that they are difficult to apply to the solution of any problem. In this study, the authors take the pragmatic perspective that, wherever appropriate, the method should aim for simplicity and focus instead of generality.

Academic and industrial experience suggests that systems modeling techniques such as the Design Structure Matrix (DSM)[8] can provide powerful tools to support design planning. However, it is important that end users remember that the real design process exhibits considerably more complexity than is captured in any model describing the abstract arrangement of its constituent tasks. To use such methods effectively, it is necessary to hold an appreciation of the limitations of the modeling framework and its relationship to the full, human complexity of the design process; in other words, users should understand the assumptions inherent to the chosen approach and be able to evaluate the appropriateness of the conclusions which may be drawn from modeling and analysis. Again, this suggests that a simple approach may be most effective in practice.

The design process should also be understood as the context into which support solutions will eventually be delivered. In a number of companies, process improvement activities which focus on mid- to long-term management goals have met with limited success, due to a failure to adequately engage with the short-term interests, motivations and requirements of other stakeholders. In the case of model-based planning methods, a critical precursor to deployment is that direct, short-term benefit must be provided to the technical design teams who are expected to support the modelling activity. The method described in this paper aims to overcome some shortcomings of previous research by adopting a requirements driven approach, in which we aim to achieve the simplest and most focused method which provides direct benefit to both management and engineering personnel. In particular, this paper focuses on the elicitation, representation and manipulation of process information; the importance of these areas were found particularly important during development of the approach.

In summary, our recent experience suggests that, in many respects and situations, the most advanced modeling and analysis techniques are no more attractive to industry than the very simplest. While we believe that advanced analytical research is of significant value [9], our focus in this study was to apply such insights to the development of a pragmatic approach for direct improvement of industrial practice. In order to provide simplicity that is useful against the background of complexity and uncertainty in the design process, we have positioned all components of the method with respect to requirements identified during the study. In the remainder of this paper, we discuss the development of the new approach under four headings: process representation; knowledge elicitation; model visualization; and model analysis.

Process representation

A number of activity-based frameworks have been proposed for modeling the design process [10]. 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 [11]. Many others are based on the DSM [8]. These may be used to capture the effects of task ordering on process effectiveness [12]; extensions capture uncertainty in task durations and outcomes [13]. The parallel iteration model proposed by AitSahlia et al [14] and the overlapping task model of Krishnan et al [15] may be used to describe processes in which the boundaries of design tasks are uncertain.

These examples all describe the design process in terms of direct relationships between tasks. Other schemes capture process behavior by explicitly capturing information flow between tasks. For example, McMahon and Xianyi [16] used Petri nets to automate repetitive crankshaft design by capturing the flows of information between computer tasks. Dynamic frameworks such as Signposting [1] and the Adaptive Test Process [17] 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 can include data files and reports as well as numerical parameterizations of the design.

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 required to carry out 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 or recommended 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 is possible if each parameter in the task’s input state is available to at least the specified level of confidence, and is recommended 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 the approach which has previously been exploited.
in process optimization applications [9]. The Signposting model of the design process is summarized in figure 2; for further detail the reader is referred to [1] and [9].

![Figure 2: The Signposting model of the design process](image)

The Signposting framework is especially appropriate in the modeling of adaptive or variant design processes; in such scenarios the majority of tasks and parameters are well delimited and may be identified in advance [18]. This is the case with the blisk design process studied in this work, which may be described to a large extent in terms of the use of design and analysis tools.

**Selection of modelling framework**

Our experience suggests that, perhaps due to the abstract nature of process-related concepts such as iteration, the forms of description available to the modeler can have a disproportionate influence on their characterization and interpretation of process behavior. In other words, the framework used to model a process can have a significant effect on the modelers thinking; in comparison to, for example, a CAD rendering of a component which obscures a critical feature, it can seem remarkably difficult to pinpoint exactly why a modeling approach does not seem to capture the behavior of a design process. It is thus of critical importance to select a framework which is appropriate to the task at hand. Further, users’ interpretations of and interactions with process descriptions according to experience and shared understanding has important consequences for the design of both the representation and visualization aspects of methods.

A key requirement for the modeling framework in this research was the need to capture tasks, parameters, and key design iterations. Early in the study a simple task/parameter matrix representation was selected to describe the blisk design process, due to the ease of manipulation of such models using commonly available spreadsheet software. However, it was later recognized that process knowledge would need to be represented within a framework that captured the context of tasks in a way which supported the flexible re-configuration of the plan of work in response to schedule slippage caused by, e.g., the discovery of unanticipated rework. This led to the adoption of the Signposting framework as the basis for process representation in this work.

In addition to the need to capture tasks, parameters, iterations and dynamic process behavior, a key factor guiding the development of the new modeling framework was the need for effective visualization of models constructed using the framework. Further, a full appreciation of the flexibility of process configuration should not be required to use the software in its simplest function – that of easily describing processes as flowchart diagrams. The following paragraphs describe minor extensions to the Signposting framework which were developed to support its use in this context.

**The Applied Signposting Model**

The Applied Signposting Model (ASM) extensions stem from two observations regarding task selection in blisk design: 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, it is believed by experienced engineers that only a small set of these possible routes are likely to occur in practice. The extensions described in the following sections are designed to support the modeling and later manipulation of this anticipated subset of routes.

**Task definitions**

ASM models are constructed hierarchically from the basic elements of tasks, parameters, processes and containers. The ASM provides three types of task to allow the modeler to easily distinguish between classes of task behavior: Simple tasks, which link one input state to one output state; Signposting tasks, which contain one input state and many output states, selected stochastically; and Iteration construct tasks, which contain one input state and exactly two output states. Simple tasks are used to capture activities whose outcome does not directly affect process routes, typically detailing or data transfer tasks. Iteration constructs are typically used to represent evaluation tasks which may result in the discovery of rework. O’Donovan [19] provides a detailed account of the use of Signposting tasks to model more complex scenarios.

In addition to capturing input and output parameters, the modeler may specify any resource requirements and estimates of the best case, most likely, and worst case duration for each task. This approach to modeling uncertainty was chosen due to its simplicity and engineers’ and managers’ familiarity with probabilistic methods.

The Applied Signposting task definitions are depicted graphically in figure 3.

![Figure 3: Applied Signposting Model task definitions](image)
**Graphical notation**

A key component of the ASM is the graphical, color-coded notation shown throughout this paper and summarized in figure 4. In this notation, ellipses are used to denote each possible combination of parameter and confidence within a process, as determined by the states defined by the tasks in the process. This allows an ASM model to be assembled from a combination of knowledge about individual tasks and, as is more appropriate in the context of this paper, from a procedural overview of process segments. Such models may then be presented and manipulated as an information flow diagram. With the software support described in following sections, this visualization allows the description of processes using an intuitive ‘flowchart’ format which conforms to the formal syntax of the ASM (sections of the blisk design model may be seen in figure 1 and 8). In comparison to a flowchart rendered in a standard office package, this formalism allows an ASM model to be “executed” using the simple algorithm described in the forthcoming analysis section.

**Hierarchical composition**

As with previous generations of the Signposting model, tasks are defined within the context of their input and output parameters. In the ASM extension, both tasks and parameters are defined within the context of a process, which is defined as a set of tasks and parameters together with a single input and single output state that are drawn from the parameters defined 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.

Hierarchical composition of process models was found to be useful for three reasons beyond simplifying the dynamic task selection as described above: firstly, in supporting designers’ interpretation of large models by providing additional contextual information for each task; secondly, in supporting the navigation of such models using interactive visualization techniques; and thirdly, in defining processes to form generic ‘building blocks’ which may be used throughout the model to represent similar areas of activity. While the first two scenarios may be supported by a simple hierarchical structuring approach in which parent-child processes are linked by direct compositional relationships, the third requires lattice structuring or indirect compositional relationships (Figure 5).

**Summary**

In summary, the ASM extensions to the Signposting framework are designed to improve simplicity in application to the graphical modeling of well-constrained dynamic processes. By capturing process knowledge purely in terms of the input and output characteristics of Signposting tasks, design may still be described as a dynamic, parameter-driven process which captures a very large set of possible routes. Combining simple tasks with iteration constructs allows the definition of a static, highly constrained process model that may be easily visualized as a flowchart. These paradigms may be combined directly, or may be separated using hierarchical composition of process segments.

**Knowledge elicitation**

Irrespective of the chosen modeling framework, the process of knowledge elicitation forms an undeniably strong...
influence on the nature of the resulting description; any process may be adequately described at many different levels of abstraction and, ultimately, the requirements, decisions and preconceptions of the modeler will determine the range of applicability of the model. While an appropriate modeling framework is a prerequisite to describing process knowledge in a manner effective for a given purpose, the knowledge elicitation and representation activities must also be informed by insight on the part of the modeler. Especially important is an appreciation of the subjectivity inherent to the elicitation of knowledge and the ambiguity of its representation within the chosen framework – an understanding of the technical details of the process itself is also required in order to develop a coherent description which is both representative and useful.

A key cause of uncertainty about process behavior may be found in the subjective nature of process knowledge and its distribution among members of a design team or company. This has important consequences for eliciting knowledge in terms of experts’ judgments of uncertainty [20] but also in terms of process description itself; in the case study described in this paper, a substantial period of time was spent attempting to synthesize a coherent perspective of the design process which was acceptable to engineers from the different disciplines involved. Furthermore, in addition to the challenge of capturing process knowledge and rationale, constructing a process model carries the additional difficulty of describing design tasks in a way which, when taken together, represent an uninterrupted route or set of routes through the process. In any elicitation exercise, the extent to which this can be achieved is strongly influenced by the expertise of the modeler, their familiarity with the framework they are using and the process they are attempting to capture.

Process knowledge may be elicited from many sources, including human experts, explicit process maps and other documentation. Consequently, no single technique may be uniformly applied; this ‘bottleneck of knowledge elicitation’ is well documented [21]. It is not obvious how the more subtle insights of knowledge elicitation and modeling may be gained except through ‘learning by doing’ – the identification, characterization and transfer of knowledge about process modeling is an important area for future research which may have the potential for disproportionate impact on the effectiveness of process modeling activities in industry.

In this study, the researcher acted as facilitator throughout the modeling process, which was driven by the engineer responsible for timely delivery of the blisk. This arrangement was influenced by the need to develop the new method and prototype software concurrently with the model elicitation activities. The blisk process model was developed incrementally throughout the study using an iterative process of critique and refinement. Expert knowledge regarding the content and requirements for the 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 modeling activity resulted in the development of a library of ‘building block’ processes consisting of around 100 tasks which, at the level of abstraction captured in the model, were used in many contexts throughout the process.

Uncertainty inherent to the process was modeled at three distinct levels: durations of certain design tasks were described using triangular probability density functions; rework cycles were specified within process blocks, representing rework arising from inadequacies discovered during data integration and validation activities; finally, the dynamic task selection features of the ASM were used to model the possibilities for arrangement of the 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.

**Model visualisation**

Model visualizations serve a number of purposes, including: aids to verbal or written communication about the process; placeholders for the modeler’s knowledge; and as a tool for knowledge elicitation. Just as a model should be developed to provide a useful abstraction of reality, so a visualization should be designed to provide a useful perspective of the underlying model. For example, the DSM is a useful technique for highlighting interdependent groups in a large, loosely coupled system of homogeneous elements [22]. It is less ideal in transferring the same information to a novice designer who wishes to know ‘which task should I attempt next?’

The prototype software developed early during this study allows three alternative visualizations of the ASM model elements: an individual task view showing input and output states together with resource and duration information; 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 7).

![Figure 7: Screenshots of early prototype software showing task state, task/parameter matrix and network visualizations from a non-hierarchical process (left - right)](image)

During the course of the study reported in this paper, the form of visualization chosen for use during knowledge elicitation and model construction was found to strongly influence the result of these activities, particularly in terms of the granularity of modeling. For example, the use of a matrix format during group elicitation sessions has been reported useful in providing structure to the exercise [23], 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 information elicited or captured purely using the task/parameter matrix or task state views resulted in a very strongly connected process. However, when presented with a flowchart visualization, the same users found ways to describe their process knowledge in a more linear fashion which resulted in a visually appealing diagram.

It was also found 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 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.

**Network layout**

The automatic layout of networks for process visualization presents a substantial research and programming challenge which lies outside the scope of this paper. In the case study, the dot algorithm distributed in the AT&T GraphViz package [24] was used to calculate such layouts, allowing the rapid development of the prototype software. Experience with this system has highlighted a number of shortcomings, most notably that the ideal layout algorithm would be concerned with more than purely aesthetic factors such as minimizing edge intersections or screen area; the perceived content of a diagram is dependent upon semiotic associations and the direct perception of relational structures [25], factors which may be influenced in subtle ways by the construction of the diagram.

To illustrate, a number of engineers commented during the study that the vertical positioning of task nodes relative to parameter nodes and the general direction of edges were more important than achieving aesthetically pleasing layouts. For example, on large layouts the dot algorithm tends to position some tasks well above some of their input parameters. In this situation, the misplaced parameter nodes were repeatedly overlooked when reading the diagram – suggesting that in addition to layout the interpretation of formal diagrams may be further influenced by the semantics, or perceived semantics, of the underlying model. Crilly et al also suggest that in diagrammatic elicitation exercises it is necessary to consider the connotations of diagrams in addition to their direct denotations [25]. Despite the importance of cognitive factors in influencing the effectiveness of network visualizations, few publications in the graph layout community address these issues [26].

Following these observations, an interactive network editor was developed to complement the task-parameter matrix method for manipulating the process information. The editor allows users to combine dot layouts with manual manipulation of the process network to achieve a pleasing diagram. Initial experimentation with this capability has further highlighted both the subtleties and importance of designing effective visualizations for the elicitation and manipulation of process information.

The importance of effective user interfaces in working with larger models and the need to experiment with interactive visualization strategies has led to the re-development of the prototype software to provide a platform upon which these issues are being explored (figure 8).

**Figure 8: Prototype developed to support development of user interface and interactive visualisation strategies**

**Model analysis**

In the following section, we describe an 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 develop a schedule based on the results of this simulation.

**Execution**

The primary requirement of the execution algorithm described in this section is to operate effectively upon an ASM process which is described using the intuitive graphical notation developed in previous sections. The dynamic task selection algorithm of the original Signposting model is not directly compatible with the simplified iteration paradigm which is used in this notation; in particular, the requirement for execution of a task to result in a direct increase in confidence stipulates that, to operate as implied by the graphical notation,
the ‘iterate again’ output state of an iteration construct task would need to reduce the level of confidence in *all* parameters ‘inside’ the defined iteration cycle.

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 modeling information flow used by McMahon [16]. 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 initialization. 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 in this scheme. Upon completion of the task, all input places are made active and all output places are made tokenized. An example step of this algorithm is depicted in figure 9.

**Figure 9: ASM model showing tri-state places (circles) & transitions (horizontal bars)**

**Process simulation**

The execution method described above allows the simulation of an ASM plan using a straightforward discrete event algorithm described in overview in figure 10. A number of authors have described methods that aim to explore or improve aspects of design practice using this type of simulation. These include the computation of probability distributions of process metrics such as risk, cost and lead time [27]; the calculation of stochastic measures of task criticality [28]; and the development of probabilistic schedules [9]. A detailed discussion of the application of simulation techniques to explore the properties of resource-constrained Signposting models is beyond the scope of this paper; for further detail the reader is referred to O’Donovan [19].

**Figure 10: Overview of the process simulation algorithm**

**Schedule development**

O’Donovan describes how scheduling information may be extracted from a population of possible processes obtained through discrete event simulation [9]. Such information may be 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 gives a prescriptive perspective; that is, which can provide answers to questions such as ‘In order to meet a certain milestone for completion of the process, when must I complete each constituent task?’ To achieve this a set of candidate processes is extracted from the population and the ‘most representative’ is extracted from this set and rendered as a conventional Gantt chart. Selecting a single process run in this way allows the depiction of detailed task information which may be directly related to the relationships between tasks captured in the original, cyclic planning network. This clear, graphical relationship, illustrated in figure 1, allows the network to be easily manipulated in order to achieve desired changes to the resulting schedules.

The simple heuristic currently used to obtain a representative set of processes based on a given milestone constraint is described graphically in figure 11. The ‘most representative’ member of this set is currently selected 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. Although this naïve approach has proved sufficient in the context of this study, it is recognized that a more advanced algorithm would be necessary before the method may successfully be applied to larger planning problems or to highly parallel processes. Previous work has resulted in the development of techniques for the optimization of design processes captured in the Signposting framework [19]; we are investigating methods by which such tools can be used for selecting a representative process run without compromising the pragmatic simplicity of our approach.

**Assessing schedule risk**

As described early in the paper, an estimation of schedule risk can be used to assess the feasibility of plans developed using our approach. The risk value displayed in figure 1 is calculated by dividing the number of ‘failed’ process runs by the total number of simulated process runs, where failed
processes are defined as those which exceed the target date (figure 11).

**Figure 11: Extracting a representative process from the population of possibilities revealed through simulation**

**EVALUATION AND FURTHER WORK**

It is difficult to define concrete success criteria for this research due to the long timescales and many influences on the effectiveness of design planning. However, at the conclusion of the study, the method had allowed the company to develop both an integrated model of their blisk design process and a plan of design activities, both of which exhibited significantly greater detail than had been achieved using the unstructured techniques previously in use. As it was believed that the primary benefit of the new method lay in its structuring of the planning activity, it is difficult to define entirely objective success criteria for this research. The most appropriate success criterion will ultimately lie in the adoption of the method by engineers and managers. At the request of personnel directly and indirectly involved with the research, the modeling/analysis tool developed as part of the study has been redeveloped and is currently undergoing further evaluation in the company.

The research described in this paper has highlighted a number of promising areas for future work. These include: the development of methods to account for the subjectivity of process knowledge elicited from many sources; the development of more effective visualization techniques which account for the cognitive factors surrounding network layout; and investigation into the appropriate use of process optimization techniques to guide the construction of plans using the new method. Finally, resource constraints typically play an important role in formulating design plans. Although the model described in this paper allows the specification of resource usage for each task, the work has not addressed the role of these constraints in determining process behavior. This is an important area for further research.

**CONCLUSIONS**

This paper has described the development of a pragmatic approach to support planning practice in industry through more detailed task planning in the complex design process. The approach has been developed to address limitations in existing planning techniques in capturing the uncertainty in activity ordering and duration which is endemic to much of engineering design. These limitations can render it difficult for engineers and managers to develop an overview of the design process and the plan of work.

The approach extends commonly available Gantt-based planning packages through the development of a modeling framework which captures tasks and their relationships to other tasks, in terms of input and output parameters and each task’s hierarchical context within the overall process. This allows the explicit specification of possible task rework scenarios in the form of cyclic parameter dependencies, and the implicit specification of parametric refinement situations through the re-use of tasks or groups of tasks in different contexts throughout the process model. Durations may be attached to each task in terms of best case, most likely and worst case estimates and each dependency cycle may be parameterized by specifying how it will be resolved, in terms of a number of planned refinement passes, iteration until a given deadline, or a probability of discovering the need for rework of earlier tasks. These estimates of the planned process route allow cyclic process models developed using the approach to be resolved into a linear sequence of tasks to be displayed as a Gantt chart.

This approach can provide several benefits when the design process is adaptive in nature, with well-delimited activities which are undertaken to drive the iterative refinement of design parameters. In such cases, we argue that a structured, model-based approach to design planning can result in improvements in the fidelity of design plans and their representations. By improving visibility of the design process, the method described in this paper can assist managers in the early identification and remedy of schedule slippages and possible process bottlenecks; ultimately, increased visibility can support risk management and reduce product development cycle time. Capturing each task in terms of its context within the process, rather than its direct dependencies upon other tasks, simplifies the reconfiguration of process information and improves maintainability of the resulting models.

This paper has focused on describing the knowledge elicitation, representation and manipulation aspects of the approach; areas which formed a bottleneck during the study. Addressing these issues was critical in gaining support of the approach within the company. In particular, the availability of suitable software tools to facilitate the process modeling proved key during the iterative process of modeling, feedback and refinement. These tools have provided significant additional benefits in enabling the integration of previously disparate process documentation within a coherent, interactive, graphical format. Although not an initial objective of the research, this direct benefit gained the interest of the technical design team who supported the modeling work during the study.

Although the modeling and analysis of design processes can provide immediate benefit to a company in the form of insights into process behavior and the identification of possible improvements, it can be difficult to achieve more sustained impact in the absence of appropriate incentives and training are not provided within the company. The approach taken by this work to address this issue is to place such tools within the context of a method which offers direct benefit to all stakeholders.
ACKNOWLEDGMENTS

This research is funded by the EPSRC. The authors would especially like to thank Neil Brown and Chris Powell for their continued support of the work, plus Tobias Brandvik and Seena Nair for their programming contributions.

REFERENCES


