INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

EVALUATION OF PROCESS MODELLING APPROACHES TO SUPPORT PROBABILISTIC DESIGN ANALYSIS

YM Goh, JD Booker and CA McMahon

Abstract

Probabilistic design analysis deals with uncertainty in engineering data. Poor correlation between analytical and experimental results suggests that improved understanding of uncertainties and knowledge of the accuracy of analysis functions is necessary to increase confidence in the probabilistic approach. It is proposed in this paper that process modelling may be used as a basis for mapping of interrelated activities in complex systems and in so doing may form a basis for representing this knowledge and understanding. In particular, uncertainties associated with the relationships between variables and the performance parameters (transfer functions) may be understood more effectively through the use of process modelling. In the paper, desirable characteristics of process modelling approaches are identified, and used to review a number of process modelling approaches and to discuss the requirements for adaptation of these models to better model probabilistic design analysis.

Keywords: Engineering analysis, knowledge management, probabilistic design, process modelling

1. Introduction

The term design analysis, as informed by [1] and [2], describes a system that transforms initial state operands (input variables such as design parameters) to final state operands (performance parameters) with the purpose of evaluating the performance of an engineering artefact. The activities (transformation nodes) in this transformation system are connected by a set of relationships called transfer functions. The transfer functions refer to mathematical and computational models developed from first principles, empirical relationships and heuristics. Approaches to engineering design analysis have traditionally been deterministic but these often result in inconsistent or sub-optimal designs owing to limited application of information about the uncertainty in the analysis variables. Many probabilistic design techniques have been developed to better account for this uncertainty, but the use of these techniques in engineering analysis has been limited owing to a number of concerns, including the heavy computational load of probabilistic analysis, the need for extensive data on uncertainties and limitations in the capability of analyses to predict the behaviour of physical systems.

Uncertainty in the variables may be characterised relatively easily. In probabilistic design, variability in the data is normally accounted for by representing geometrical parameters, loading history and material properties with probability distributions. However, while knowledge of variability is available in well-established design domains, uncertainty in transfer functions and factors affecting it are not always well understood [3], as observed in the inconsistency between analytical results and experimental measurements [4] [5]. Uncertainties associated with the transfer functions could arise from conceptual or modelling errors, lack of information and systematic errors in data acquisition and data analysis (e.g.

sampling and curve fitting) [6]. These limitations have contributed to the lack of confidence in probabilistic design and inhibit its application. Some critical engineering domains such as the automotive and aerospace still rely heavily on prototype testing for analysis and validation. Evidently, uncertainty in these analytical models is an issue that needs more emphasis in accumulation of knowledge and understanding.

Design analysis can rarely be characterised as a single activity. It is more likely to consist of a series of complex transformation activities described by a number of local transfer functions, where information flows from one activity to another. To quantify how uncertainty propagates through analytical models, a systematic way of collecting information based on formal process modelling is proposed to identify uncertainty in transfer functions and the sensitivity towards uncertainty in the data. An example of a highly complex engineering design analysis is the modelling of the development of residual stresses in a manufacturing process such as forging followed by air cooling. This process can be modelled as a composition of a set of thermal and mechanical analyses, where the transfer functions involved are highly implicit, nonlinear and dynamic, typically solved using computational methods e.g. finite element analysis. Uncertainties associated with these transfer functions may be difficult to characterise, but this may be feasible when knowledge is accumulated in a systematic manner regarding the variables, transfer functions and the validity of the assumptions and approximations made in the analytical procedures. It is suggested that using process modelling as a framework may provide a suitable basis for such a systematic approach. This paper outlines the requirements for modelling of probabilistic analyses using process modelling and then reviews seven process modelling approaches against these requirements.

2. Modelling requirements

Models of interrelated or sequential activities are constructed to study and understand complex systems and to facilitate the visualisation of information flow in the systems. Various modelling approaches have been applied extensively in software development, and in modelling such domains as business systems and manufacturing systems. The modelling of activities and information flows in engineering design processes has also been done widely through various process modelling techniques [7]. In this section, important aspects of probabilistic design analysis modelling are discussed. The primary concern is to identify an approach that allows important information about an analysis process to be modelled; a secondary concern is to create a dynamic, executable model of a process. The human systems and management aspects are of less importance here. With this purpose in mind, requirements for effective modelling of probabilistic design analysis are:

The model should allow definition of uncertainty in information and activities

The process models need to model uncertainties associated with the design parameters such as basic variables in transfer functions. In probabilistic design, these are non-deterministic, therefore the process model must be able to support the input, propagation and output of uncertain variables. This suggests the need for a separate definition of data entities, where uncertainty definition can be associated with these entities explicitly.

The model should support prioritisation of efforts

As noted previously, uncertainties are introduced at each transformation node (i.e. transfer function) in the analytical model. Some of the transfer functions at these nodes are not well understood or may be difficult to characterise. An effective process model must be able to describe the information uncertainty, incompleteness and confidence level associated with each transfer function node in the model to highlight activities that are not well understood or in which there is low confidence. Thus, it requires explicit representation of uncertainties in each node to support prioritisation of efforts in understanding and accumulation of knowledge.

The model should allow a building block approach of a detailed process.

Owing to the complexity of design analyses, the process model should facilitate the description of a system at different levels of detail, where an object definition is refined incrementally at each level. In modelling a probabilistic design analysis, a building block approach is appropriate to systematically represent the sequence of activities in the analytical model and to provide a formal modelling structure, which enables explicit identification and capture of information from the model.

The model should allow modelling of the states of a system.

States of a system are instantaneous snapshots of the entire system at a specific point in time. In design analysis, the network in the process model needs to reorganise (update as it progresses) to represent the temporal state of a system as it evolves with time. This time dependent behaviour is particularly important for simulation purposes and it enables an executable representation of a dynamic system.

The model should incorporate conditional elements.

The purpose of the construction of a process model is to aggregate uncertainties introduced in each transformation node in design analysis. There are many ways of carrying out the design analysis, and it may be performed in different ways under different circumstances depending on many factors or constraints. Conditional elements allow variation of the transformation activities in the process model according to these circumstances.

3. Review of methods in process modelling

It is noted that the methods discussed here are not exhaustive, but selected from techniques that are typically available to designers for modelling design processes. These approaches are reviewed in brief in this section and are grouped according to four modelling views of a system, namely the functional, dynamic, object and task-based views.

3.1 Functional modelling

Functional models describe data flow and transformation in a system. The graphical notations typically used are nodes to represent processes and arcs to represent data flows. Three types of process modelling approaches that belong to this category are the data flow diagram, the structured systems analysis and design method and the Integrated Definition IDEF0 method.

Data Flow Diagram (DFD)

First introduced in 1970s, DFD is one of the oldest and simplest modelling techniques which is still widely used today. Each node in DFD represents a process or activity in which data is processed, thus modelling the information flows in a system. It has various graphical notations to represent the process, external agent, data store and data flow. The basic notations are illustrated in Figure 1 (a). As a functional model, DFD allows explicit identification of the transformation nodes (or activities) in a system but it only indicates the direction of information flow in the process. As noted earlier, separate definition of data entities is needed to allow for uncertainty definition in the design parameters in probabilistic design analysis, which is not present in DFD. In addition, DFD does not incorporate a time element, therefore it is not capable of modelling dynamic processes.

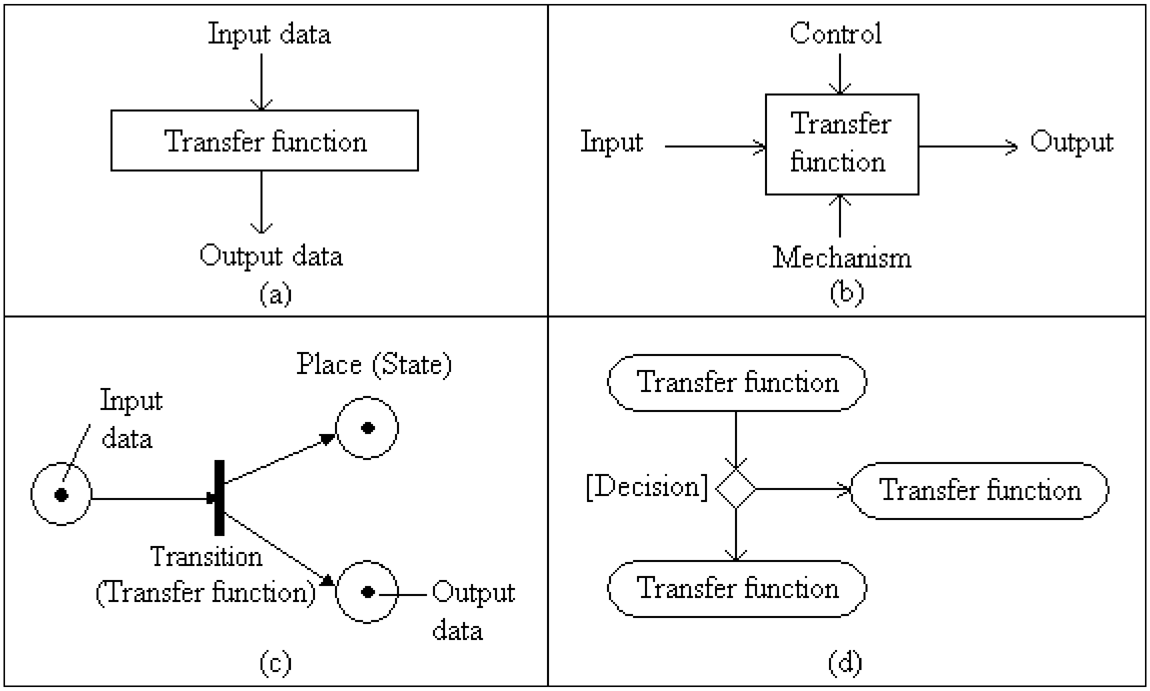

Figure 1: Basic notations in (a) DFD and (b) IDEF0 (c) Petri net (d) UML activity diagram.

Structured Systems Analysis and Design Method (SSADM)

Introduced in the 1980s, SSADM [8] [9] was developed from DFD with additional views for Logical Data Structures (LDS) and Entity Life Histories (ELH) to enhance the modelling capabilities. SSADM is used in the design and analysis stages of systems. DFD, as before, is used to model transformation processes together with data flows in a system. Of the additional views in SSADM, LDS is employed to give a structural view of system data and ELH to incorporate the effect of time on the system data, thus allowing for modelling of the dynamic evolution of a system.

Integrated Definition (IDEF0)

Developed by the US Air Force, the IDEF family consists of a series of methods that model different views of a system. For instance, IDEF1 is an information model, IDEF2 is a dynamic model and IDEF3 is a process description model. IDEF0 [10], one of the most commonly used, is a functional model and is derived from Structured Analysis and Design Technique (SADT). For conciseness, only IDEF0 is reviewed here.

IDEF0 is used to produce a structured function model to gain understanding, support analysis, provide logic for potential changes, specify requirements, or support systems level design and integration activities. An IDEF0 model is composed of a set of hierarchically linked diagrams with supporting text that display increasing levels of detail describing functions and their interfaces within the context of a system. The model describes what a system does, what controls it, what things it works on, what means it uses to perform its functions, and what it produces. The context, viewpoint and purpose of the model define its orientation and should be fixed before the model is created.

The four components in the IDEF0 notation, as shown in Figure 1 (b) are Inputs (I), Controls (C), Outputs (O) and Mechanisms (M), collectively termed ICOM. Input data or objects are transformed by the function to produce the output, which is the result of a transformation. A control is a component that is used as guidance and not consumed or transformed by the process or activity, e.g. standards, policies, guidelines etc. The mechanism is the means used to perform a function, e.g. people, manual tools, automated tools etc. Mechanisms and controls are also present in reference [2], identified as tactical instruments (resources), operators and the means of making available appropriate knowledge and information. IDEF0 models the transformation nodes in probabilistic design analysis well, but like DFD, does not allow for explicit definition of data entities. It is also difficult to incorporate into simulation as it does not allow for dynamic evolution of the network. IDEF0 gives a static view of a system.

3.2 Dynamic modelling

Dynamic models describe the interactions among objects and the time dependent behaviour of a system. They model the sequence of activities in a system, thus allowing control of the system. A dynamic model consists of nodes to represent states of a system and arcs to represent transitions between states. A typical model is a state transition diagram.

Petri Net (with classical, colour, time and hierarchy variants)

Petri nets are a dynamic modelling approach that were developed from state transition diagrams. They comprise two types of node: place and transition, as shown in Figure 1 (c) [11]. A transition in Petri net is equivalent to a transformation node described by a transfer function. Directed arcs connect transitions and places to indicate the flow of information and indicate the sequential relationship between the nodes. A classical Petri net consists of a four-tuple <P, T, I, O> corresponding to Places (P), Transitions (T), Inputs (I) and Outputs (O) respectively with an initial marking, μ. Marking is shown by places with a number of tokens represented by small dots.

An evolution of state corresponds to an evolution of the marking, caused by firing of a transition. A transition fires by removing tokens from its input places and creating new tokens which are distributed to its output places. This activity transforms the data from its input state to its output state. A transition can only be fired if each of the input places contains at least one token, the transition is said to be enabled. Petri net could model the processes included in the system, the relationship, data exchange and sharing and the dynamic state of the process. Extensions to classical Petri nets are:

- Using colour to model data.

  A coloured Petri net (CPN) [12] is a special case of a Petri net in which the tokens have identifying attributes. They allow the use of tokens that carry data values and can hence be distinguished from one another. A CPN is a five-tuple <P, T, C, I, O>, where C is a colour function. CPN allows definition and manipulation of data values. This is a required property to support probabilistic design analysis, as data that propagate through the network contain uncertainty information.

- Modelling dynamic processes.

  The time dependent nature of a system is described by assigning time to the activities, so that duration and delays could be modelled. The time concept can be introduced in timed Petri nets via attributes of the tokens, places or transitions. This extension of the classical Petri net further supports dynamic state modelling, allowing the net to model the time dependent behaviour of a system.

- Modelling of large systems.

  In a hierarchical Petri net, the model can be extended on a separate net called a subnet. A subnet is an aggregate of a number of places, transitions, and subsystems. Large Petri nets can be broken down to several hierarchies of Petri nets with different level of detail. This provides a method for systematic modelling of a complex system.

3.3 Object modelling

Object models describe the static, structural and data aspects of the objects in the system and the relationships between them. An object model has nodes to represent object classes and arcs to represent relationships among classes.

Unified Modelling Language (UML)

The UML [13] is a language for specifying, visualising, constructing, and documenting the artefacts of software systems, and has been extended for business modelling and other non-software systems. UML has specific notations and the related grammatical rules for constructing object-oriented models. Announced on 1 September 1997, UML unifies Booch's Object-Oriented Design, Rumbaugh's Object Modelling Technique and Jacobson's Object-Oriented Software Engineering [14]. Besides object modelling, UML also brings together other modelling aspects such as functional and dynamic views discussed previously. It consists of a series of nine diagrams to capture the static, use case, behaviour, interaction and implementation views of a system. The diagrams considered suitable to support information modelling in probabilistic design are the activity diagram and its business process extensions.

- Activity diagram.

  Activity diagrams model the behaviour view of a system. They are used to show how different processes in a system are constructed, how they start, decision paths that can be taken from start to finish and where concurrent activities may occur during execution. They are capable of modelling the information flows in the system and have conditional elements incorporated to allow for different activity paths. The basic notations in a UML activity diagram are illustrated in Figure 1 (d).

- Business process model [15].

  As an extension to the activity diagram, a business process model shows the goal of a process, and the inputs, outputs, events and information that are involved in the business process. The notation implies a flow of activities from left to right, typically an event element is placed to the left of the process and the output to the right. Both the activity diagram and its business process variant adopt a building block approach.

3.4 Task-based modelling

Recently, the engineering community has developed process modelling approaches to suit some specific needs in modelling engineering design processes. Developed specifically for task-based modelling, they are mostly concerned with task sequencing and optimising routes through the design process. Two of these techniques are discussed below.

Design Structure Matrix (DSM)

A DSM [16] is a compact, matrix representation of a system or project which highlights issues of information needs and requirements, task sequencing and iterations by specifying the dependencies or interactions between activities. A DSM matrix is then partitioned (reordered) so that the new arrangement contains the minimum feedback marks. Tearing is the process of removing a set of feedback marks from the matrix, allowing further partitioning to transform it as closely as possible to a lower triangular form. This procedure ensures fewer system elements involved in the iteration cycles, resulting in an optimal process execution.

A binary DSM is typically populated with 1 and 0 in each matrix cell to signify the existence or absence of a dependency between elements of a system respectively. A numerical DSM (NDSM) [17], is an extension to the binary DSM in which cells may model different attributes that provide more detailed information on the relationships between the different activities. It provides a better understanding of the system and allows for the development of more complex and practical partitioning and tearing. There are several types of NDSM depending on the data type modelled, including task-based, parameter-based, team-based and component-based NDSMs. A task based NDSM allows for prioritisation effort as confidence levels may be assigned to activities in the system whereas a parameter-based allows for definition of uncertainty in the design parameters.

Signposting

The Signposting tool [18] was developed based on the assumption that the design process may be constructed from a predefined set of tasks. The key parameters in a task are then identified, supported by a knowledge model, in which the confidence in these parameters is used to prioritise or “signpost” the next appropriate task. Signposting contains information regarding the relative importance of tasks, a confidence mapping to indicate new parameters produced and confidence changes in existing parameters. The confidence matrix is used to relate the minimum confidence of the input parameters required to give a particular level of confidence in the output parameters using information available. Signposting incorporates issues of prioritisation of tasks based on confidence associated with them.

4. Results and discussions

Results of the assessment of each process modelling approach against the modelling requirements outlined in Section 2 for uncertainty quantification in probabilistic design analysis are summarised in Table 1. This illustrates the requirement satisfied by each process model (together with notes where necessary) and thus identifies which may form the basis for the development of an adapted process modelling approach to suit the proposed work to incorporate uncertainty modelling in probabilistic design analysis.

Table 1: Process modelling approaches and the requirements for uncertainty quantification

Modelling approaches	DFD	SSADM	IDEF0	Petri Net				UML	DSM	Signposting
Extensions				Classical	Coloured[1]	Time[1]	Hierarchy[1]			
Definition of uncertainty	✗	✗	✗	✗	✓	✗	✗	✓[2]	✓[3]	✓[4]
Supports prioritisation	✗	✗	✗	✗	✗	✗	✗	✗	✓[3]	✓
Building block	✓	✓	✓	✗	✗	✗	✓	O[5]	✗	✗
States of a system	✗	✓[6]	✗[7]	✓	✓	✓	✓	✗[8]	✗	✓[4]
Conditional elements	✗	✗	✗[7]	O[9]	O[9]	O[9]	O[9]	✓	✗	✗

✓ Yes O Yes but not clear ✗ No

Notes to Table 1:

1. These are extensions to classical Petri net and may be used in any combination with the classical Petri net.
2. Uncertainty can be modelled in object diagrams.
3. Prioritisation of efforts can be achieved via a task based NDSM. Uncertainty definition can be achieved via a parameter based NDSM.
4. Signposting includes states and uncertainties in the objects/information using symbols to represent different confidence levels.
5. A block of activity or action state in the UML activity diagram notation represent different levels of granularity, therefore it is not an obvious building block approach. Business process model is a building block approach.
6. With ELH view.
7. States of a system can be modelled with IDEF3. Conditional elements are present in IDEF3.
8. Not possible with business process and activity diagram but possible with a UML statechart, which is a dynamic model. The statechart is not reviewed in this paper.
9. Conditional elements are not present in Petri net models, however conditions can be associated with arc expressions, or by manipulating the marking of the net.

Table 2: Process modelling approaches and their attributes to support probabilistic design.

Modelling approaches	IDEF0	Petri Net			UML – Activity diagram
Extensions		Classical	Coloured & Time	Hierarchy	
Modelling level of detail: Level of detail	Multiple	Low level	Low level	Multiple	Multiple
Multiple abstractions	✓	✗	✗	✓	✓
User issues: Dependence on user experience	Highly dependent	Less dependent	Less dependent	Less dependent	Less dependent
Complexity of notation	Simple	Simple	Simple	Simple	Moderate
Computer supported tools	✓	✓	✓	✓	✓
Availability of standards	US FIPS PUB 183	ISO/IEC 15909	ISO/IEC 15909	ISO/IEC 15909	ISO/IEC DIS19501
How long used?	30 years	40 years	20 years	12 years	5 years
Scope of modelling: Measure of conciseness	Orderly but elaborate	Untidy network	Untidy network	Elaborate	Orderly
Modelling different views?	✓	✗	✗	✗	✓
Engineering analysis modelling: Deterministic data	✓	✓	✓	✓	✓
Probabilistic data	✗	✗	✓	✗	✗
Possibilistic data	✗	✗	✓	✗	✗
Qualitative data	✓	✗	✓	✗	✓

From the review of the process modelling approaches, some techniques may be eliminated because they are only capable of describing the process flow and optimising the design process routes. These are DFD, SSADM, DSM and Signposting. The methods considered suitable for further adaptation development to model probabilistic design analysis are IDEF0, Petri net and UML. In addition to the base requirements, the attributes of each process modelling approach to support a procedural probabilistic design analysis are also assessed. These attributes include the level of detail modelled, user issues and the scope of the modelling. These attributes are elaborated in Table 2 for IDEF0, Petri net and UML, with supporting information where appropriate.

As functional models, DFD, SSADM and IDEF0 model the transformation activities in design analysis well. However, they are insufficient to model the uncertainty in data effectively. Petri net is capable of capturing various modelling aspects with its colour, time and hierarchical extensions. The main advantage of a Petri net is its network ability to dynamically evolve, providing an executable system representation. UML allows for class definition with its object modelling approach, but the network is static and reorganisation is difficult. Various attributes of the nine types of UML diagrams may form a basis for better adaptation to uncertainty modelling. DSM and Signposting, being specially developed for design task sequencing, fall short in supporting probabilistic design analysis.

5. Conclusion

This paper has proposed the use of process modelling to map a systematic sequence of uncertain analytical activities involved in performing probabilistic design analysis. Process models are used to facilitate high level representations of analytical processes prior to the application of detailed probabilistic design, such that the relationships between inputs and outputs, the states and flow of data from one activity to the next could be well understood. For example, in the analyses of complex processes such as thermo-mechanical behaviour of materials in manufacturing processes, process models could indicate key problem areas through simple sensitivity measures. This paper then outlined the modelling requirement of a process model to suit its application to support uncertainty quantification in probabilistic design analysis. It is advocated that certain attributes of existing process modelling approaches could achieve this, however, there is no one technique that satisfies all the required criteria. The assessment of attributes for supporting a procedural probabilistic design has also highlighted several important issues that form the basis for the selection of a process modelling approach that could be best adapted for probabilistic design process to achieve a synergistic output. Used in isolation, probabilistic design and process modelling will not achieve the required uncertainty characterisation, but together, they help in the understanding of the data, the transfer functions and their interactions. It is suggested that an adapted tool that satisfies the modelling requirements outlined could lead to better uncertainty modelling. Methods found useful for adaptation are IDEF0, Petri net and UML. Future work is to develop a hybrid method based on some of the attributes of these approaches to support probabilistic design analysis with more cases of varying complexity.

References

[1] Hubka, V., and Eder, W. E., "Design Science: Introduction to the Needs, Scope and Organization of Engineering Design Knowledge", Springer-Verlag London Limited, Great Britain, 1996.

[2] Hubka, V., and Eder, W. E., "Theory of Technical Systems: A total concept theory for engineering design", Springer-Verlag Berlin Heidelberg, Germany, 1988.

[3] Jensen, H. A., and Sepulveda, A. E., "Optimal design of uncertain systems under stochastic excitation", AIAA journal, Vol. 38, 2000, pp. 2133-2141.

[4] Oberkampf, W. L., DeLand, S. M., Rutherford, B. M., Diegert, K. V., and Alvin, K. F., "Error and Uncertainty in Modelling and Simulation", Reliability Engineering and System Safety, Vol. 75, 2002, pp. 333-357.

[5] Wojtkiewicz, S. F., Eldred, M. S., Field, Jr., R.V., Urbina, A. and Red-Horse, J.R., "A Toolkit For Uncertainty Quantification In Large Computational Engineering Models," Proceedings of the 42nd AIAA SDM Conference, Seattle, Washington, 2001.

[6] Muste, M., "Sources of bias errors in flume experiments on suspended-sediment transport", Journal of Hydraulic Research, Vol. 40, 2002, pp. 695-708.

[7] Pavkovic, N., Marjanovic, D., and Dekovic, D., "Object-oriented modelling of the design process", Proceedings of ICED 01, Glasgow, 2001.

[8] Middleton, P., and McCollum, B, "Management of process improvement by prescription", Journal of Systems and Software, Vol. 57, 2001, pp. 9-19.

[9] Richardson, J., Ormerod, T. C., and Shepherd, A., "The role of task analysis in capturing requirements for interface design", Interacting with Computers, Vol. 9, 1998, pp. 367-384.

[10] Draft Federal Information Processing Standards Publication 183, "Announcing the Standard for Integration Definition For Function Modeling (IDEF0)", National Institute of Standards and Technology, 1993.

[11] Huang, Y., Jeng, M., And Chung, S., "Design, analysis and implementation of a real-world manufacturing cell controller based on Petri nets", International Journal of Computer Integrated Manufacturing, Vol. 14, 2001, pp. 304-318.

[12] Jensen, K., "Brief Introduction to Coloured Petri Nets", Proceedings of TACAS'97 Workshop, Enschede, The Netherlands, 1997.

[13] Booch, G., Rumbaugh, J., and Jacobson, I., "The Unified Modeling Language User Guide", Addison Wesley Longman, USA, 1999.

[14] Object Management Group, "OMG Unified Modeling Language Specification, Version 1.4", 2001. http://www.omg.org/technology/documents/formal/uml.htm

[15] Eriksson, H. E., and Penker, M, "Business Modelling with UML: Business patterns and business objects", John Wiley & Sons, New York, 2000.

[16] The MIT Design Structure Matrix – DSM Homepage, Massachusetts Institute of Technology, January 2001. http://web.mit.edu/dsm

[17] Yassine, A. A., Whitney, D. E., and Zambito, T., "Assessment of Rework Probabilities for Simulating Product Development Processes using the Design Structure Matrix (DSM)", Proceedings of DETC'01, Pittsburgh, Pennsylvania, 2001.

[18] Clarkson, P. J., and Hamilton, J. R., "'Signposting', A Parameter-driven Task-based Model of the Design Process", Research in Engineering Design, Vol. 12, 2000, pp. 18-38.

Dr JD Booker, University of Bristol, Department of Mechanical Engineering, Queen's Building, University Walk, Bristol BS8 1TR, UK. Tel: 01179289790 Fax: 01179294423 Email: J.D.Booker@bris.ac.uk