On the Functionality of Cambridge Advanced Modeller

A Systematic Qualitative-Quantitative Approach

Fatemeh Barzegar

Department of Management and Production Engineering

Supervisor: Prof Marco CANTAMESSA (Politecnico di Torino)

Co-supervisor: Prof P. John CLARKSON (University of Cambridge)

Advisor: Dr Mohammad HASSANNEZHAD (University of Cambridge)

December 2018

A dissertation submitted for the Degree of Master of Science in Management Engineering, Politecnico di Torino

Declaration

This work is the result of six months full-time and one-year part-time research at the Engineering Design Centre (EDC) of Cambridge University, starting from October 2016 to the end of March 2018. The confidentiality of data that used at the Cambridge EDC for the purpose of this research is reserved for the institution. Unless otherwise stated, this report is the result of my own research and does not include the outcome of work done in collaboration. Any reference to the work of others is clearly indicated in the text. This report has not been submitted in whole or in part for consideration for any other degree or qualification at this University or any other Institution.

Fatemeh Barzegar
December 2018

Acknowledgements

My first and foremost thank you goes to Prof P. John Clarkson for giving me this wonderful opportunity to work in a highly challenging and dynamic environment, and for supervising and encouraging this research. With no hesitation, this report could not be accomplished without his continuous support and constructive feedback. I would also like to thank Prof Marco Cantamessa for supervising this research and for his valuable feedback in regular meetings.

Furthermore, I would like to thank the Process and Change Management teams in Cambridge EDC for their contribution during the qualitative study, especially Nigel Ball and Seena Nair for constructive discussions and encouragements. I would also like Andrew Flintham for his valuable IT support in getting access to the CAM users' data, and Mari Huhtala and Anna Walczyk for their assistance and administrative support.

My last and principle acknowledgement is due to whom I shall be forever indebted, my wonderful parents for their endless support and encouragements, and to Dr Mohammad Hassannezhad, my husband at life and my advisor at work, for his continuous support throughout this report and patience during tough times.

Abstract

The modelling, simulation, and analysis of engineering design processes is non-trivial and understanding the behaviour of such systems is particularly challenging both for the research community and for practitioners. In order to respond to these needs, *Cambridge Advanced Modeller* (*CAM*) has been in development at *Cambridge Engineering Design Centre* (*EDC*) since 2007. Over the time, it has been well recognised as an effective software tool for modelling and analysing the project tasks and dependencies in such complex systems and been used in so many worldwide applications in both academic and commercial contexts.

However, as of its first release in 2007, and after around a decade, there has not been a generic investigation on how previously developed models in Cambridge EDC came up with challenges raising from design complexity and uncertainty, and to what extent, the CAM software has been capable to address these challenges, with regards to its multiple toolboxes. In order to address the above objectives, this report contributes to investigating the *functionality* of CAM in modelling complex engineering processes, through a systematic methodology. The general objective is (1) to enhance the capabilities of CAM and broaden its utility in supporting worldwide users on the one hand, and (2) to identify the key modelling challenges so that can be addressed in future model developments, aiming to answer: how to do effective modelling?

A systematic methodology was undertaken to address the above objectives: a hybrid *qualitative* and *quantitative* procedure. As far as related to the qualitative study, a conceptual framework developed (based on which a number of (mostly internal) users) was selected for interview and followed by an expanded survey that conducted to understand the current practice of CAM in supporting the broad range of its internal and external users.

The quantitative aspect of this study was mainly concerned with rebuilding multiple versions of well-known *Signposting* systems (that was originally developed in the same centre in 2000 and was seen several improvements over the years) in CAM. An old case of a Mechanical Design System used to re-build, simulate, and run the models. Accordingly, a range of advanced diagramming tools (such as Parallel Coordinates Plots, Dependency Matrices, and comparative (probabilistic) Gantt Charts) presented to visualise the results and followed by an expanded sensitivity analysis for performance evaluation.

Based on the observations of qualitative study (interviews and survey) and the outcomes of modelling in CAM, the supplementary discussion presented to answer the research objectives, for example, what sort of functions should be added to the software, or what modelling issues should be particularly addressed in future modelling efforts. Finally, the report concluded with directions for future researchers.

Table of Contents

List of Abbreviations

Term	Description
ABS	Actor-Based Signposting
ASM	Applied Signposting Model
ATP	Adaptive Test Process
CAM	Cambridge Advanced Modeller
CPM	Change Propagation Method
DMM	Domain Mapping matrix
DRed	Decision Rationale Editor
DSM	Dependency Structure Matrix
DP	Design Process
EDC	Engineering Design Centre
ESM	Extended Signposting Model
EPA	Exploring Possible Architectures
GERT	Graphical Evaluation and Review Technique
MDM	Multiple Domain Matrix
OSM	Original Signposting Model
PCA	Principle Component Analysis
PCD	Parallel Coordinate Diagram
PDP	Product Development Process
PERT	Project Evaluation and Review Technique
TPDF	Triangular Probability Density Function
UML	Unified Modelling Language

Chapter 1

1. Introduction

This research looks at simulation-based modelling of engineering design processes, with respect to the reusability of the developed models. Central to this research is the analysis of the functionality of CAM. It is one of the most widely used and non-commercial software platforms that have received great attention in academia and industry over the past decade, due to its capability in detailed and dynamic modelling and analysis of tasks, dependencies, and their associated changes. This chapter introduces the outline and objectives of the work and the structure based on which I aim to achieve the research objectives.

1.1. Theoretical foundation

Prior to this research, the importance of simulation-based modelling has been well recognised in the successful management of complex engineering design projects. Many studies conducted, from different perspectives, to address inherent design characteristics such as aspects of novelty, iteration, and uncertainty. Accordingly, many models developed to understand and improve multiple phases of product development process (PDP) considering its characteristics (for overview of the existing models, see Wynn and Clarkson (2017) and Browning and Ramasesh (2007) for example). *Model* in this context refers to an abstraction of a perceived or envisaged situation (Wynn and Clarkson, 2018), in which the form of the model is influenced by the intentions of the modeller (Browning et al., 2006).

By looking at the extensive literature of PDPs, addressing aspects of information dependency has been one of key challenges facing the modelling of such processes and each of the existed models has somehow been concerned with mapping *precedence* and *dependency* relationships among tasks. Popular approaches in graphical-based class are PERT, GERT, Petri-Nets, UML, ASM, and System Dynamics (focussing on modelling precedence) and in matrix-based class are the DSM techniques including DMM, MDM, and advances such as Change Prediction Method – CPM (focusing on modelling dependencies). There is another class of models relying on the *adaptive* selection of the design tasks. The popular examples are Signposting systems, Decision-based Design, Adaptive Process (ATP), and some classes of Agent-based models.

There are nonetheless valuable reviews in the literature that can cover different aspects of modelling PDPs, such as Wynn and Clarkson (2005) and (2018) Browning & Ramasesh (2007), and Browning (2016) on process modelling, Wynn and Eckert (2017) on aspects of iterations, Gerwin and Barrowman (2002) on Integrated Product Development (IPD), Krishnan and Ulrich (2001) on aspects of decisions, Jarratt et al. (2011) and Hamraz et al. (2013) on aspects of changes, Ramdas (2003) on product architecture, Heisig et al. (2014) on aspects of information processing, Eisenbart et al. (2013) on aspects of functional modelling, and Eckert et al. (2017) on integration of product and process models, among many others.

As long as advancements in developing design process (DP) models, their functionality – practical utility in supporting managers' decisions in complex design projects – has been a challenging issue. In spite of acceptable capability in formulating the problem, application of most of the existing models is confined to a single discipline (Hassannezhad, 2015, p.48) and models that are capable of incorporating multiple disciplines are usually represented at a more abstract management level (Gericke and Blessing, 2012). According to Little (1970), useful process models from the perspective of managers and decision-makers should be simple, robust, easy to control, adaptive, complete on important issues, and easy to communicate with. In this sense, reusability of a model is a critical issue in design, development, and implementation of a process model. In fact, development and application of the models to the real-life complex problems often requires specialised computer software suitable for manipulating large data sets (Wynn, Wyatt, et al., 2010).

However, there is a huge gap in the literature in terms of developing multi-purpose software platforms that can apply to a range of complex problems and support a range of managers in dealing with different project situations. In the study of Hassannezhad (2015), the author looked at the methodology of 52 recent process models in the field of PDP, in that only 17 of the models (around one-third) was come up with a user-interface. The rest was mainly used general-purpose programming languages for formulating the model or established their models on an existing (commercial) simulation platform.

In this way, and as a response to the needs of research community and practitioners, Cambridge EDC has developed a software platform for modelling, simulating, and visualising aspects of PDP complexity, termed as *Cambridge Advanced Modeller - CAM*. The reports show that, as of its original release in 2007, it has been downloaded over 5500 times by 3500 unique users around the world both from academia and world-class companies.
What makes CAM distinctive with respect to the other modelling software is that: first of all, it has always been free of charge for doing research, without any difficulty in completing the registration. Then, it has been the only software that covers multiple aspects of modelling precedence and dependencies, change propagations, and discrete-event modelling and simulation of adaptive processes in a single framework. Third, the software has continuously been updated according to the recent findings from undergoing research projects in Cambridge EDC. These issues altogether have given CAM an acceptable functionality with regard to its almost no cost.
Comparing to the other non-commercial and research-based software, CAM can give much more to a user. Its powerful mathematical engine provides enough flexibility in using the existed functions or creating a new set of functions. Accordingly, its graphical interface, while being simple to use and easy to understand for non-experts, is largely updateable and scalable so that be applied to any size of the problem. Furthermore, the software focuses on state of the art in challenges facing modelling, simulation, and visualisation of complex engineering/business processes.
In order to maintain its efficiency in solving real-life complex problems, it is necessary to understand the improvement points of the software. This can be achieved by learning from the feedback that is received from internal or external users and also by evaluating what can and cannot the software do in terms of modelling different design situations. Eventually, the improvement points will be applied to the future software updates. This requires a deep investigation on the software and is the ultimate purpose of this study.
This report, therefore, contributes to investigating the functionality of CAM software from the perspectives of user and modeller (developer). To address the former users' perspective, a range of interviews and surveys has been accomplished and the results discussed in the group meetings with the software development team. From the modeller's perspective, three different versions of the Signposting system selected while each one of which looking at the same problem from a different perspective. Accordingly, the process modelling

toolbox of CAM – the ASM toolbox (see Chapter 2.1) selected to simulate and analyse the sensitivity of the models comparatively. Apart from these main contributions, this report can be a source of knowledge for potential users of CAM whom are interested to understand the history of underlying research that has been undertaken to develop and maintain the software over the past two decades.

1.2. Research objectives

Based on the issues discussed in the theoretical foundation, the principal research question to be answered in this research is pertaining to such advancements in process modelling and simulation:

Principle research question: How can enhancing the functionality of the CAM software support modelling and analysis of complex design processes?

To answer this principle research question, this research sets out by first understanding the main and additional modelling toolboxes in CAM. For this purpose, a complete review of the literature related to both the CAM toolboxes and the research underlying each toolbox was performed to (1) understand the relative capability of CAM toolboxes in modelling different project situations and (2) understand the relative capability of previously developed models in using CAM for modelling and simulation analysis. Therefore, one more specific question raises pertaining to the understanding of the characteristics that should be investigated throughout the report to evaluate CAM:

Specific RQ1: What are the key characteristics of the CAM software (what aspects of the software) that should be considered to evaluate CAM?

Answering the first specific question (RQ1) can provide a baseline for analysing the functionality of the software with respect to its (potential) internal and external users. At the same time, it can raise new questions, relating to finding the actual users of the model – the active users whom have sufficient knowledge on functionality of CAM toolboxes –

that should be shortlisted for interview and survey, and to what extent they have been concerned with different toolboxes in CAM. Therefore, the next step is to identify the list of active users based on the records that existed in the group.

However, downloading the software does not necessarily mean that the downloader has used the software as well. Therefore, the presumption here is that the people who downloaded different versions of the software in subsequent years have had more interest in using it and consequently have more knowledge of the software so far. As the result, there existed two more specific questions to understand the users' feedback on the potential strengths and limitations of the software (in addressing the user's specific problem).

Specific RQ2: What aspects of the CAM software have been more popular between the internal and external users, with respect to the different contexts such as in academia and in industry?

Specific RQ3: How can the feedback received from interviews and surveys be used to improve functionality of future CAM updates, through analysing the data?

Apart from users' perspective, there has been a continuous interest among CAM developers (and generally, at the Cambridge EDC) on linking the advances in research (in process modelling and simulation) with the functionality of the associated toolboxes in the software. Some of these advances have been applied in the main toolboxes, while some other resulted into the development of a range of additional toolboxes and plug-ins for CAM.

Unlike the users' perspective, linking the advances in undergoing research in EDC with the CAM toolboxes requires a deep understanding of the functionality of any single icon in the software, which accordingly requires an individual goes through all these functions one-by-one. In doing so, a well-recognized model in the research community selected – so as referred to *Signposting* – which has originally been developed at the Cambridge EDC in 1999, and three of its previously developed versions comparatively modelled in order to answer the following specific questions:

Specific RQ4: what are the basic requirements of an advanced modelling software, for building, simulating, and visualising the outcomes in a user-friendly manner?

Specific RQ5: What are the key process modelling characteristics that should be reflected in future modelling attempts?

In summary, the present research starts with exploring the underlying research that constitutes the development and maintenance of the CAM toolboxes (literature review in Chapter 2). It follows by understanding the voice of internal and external users who have been used CAM to model different design and development situations (purposes) and who used different sets of functions in different toolboxes to achieve their goal (qualitative study in Chapter 3). The research then continues with comparing three different versions of Signposting system to understand the hidden challenges facing the modelling and simulation of complex design processes (quantitative study in Chapter 4). Nevertheless, the ultimate objective of this research is to help future research studies (in the area of process modelling and simulation) to provide a better representation of the reality of complex design projects. The belief is that enhancing the functionality of CAM software can be a significant step towards this goal, while can also result in widening the range of CAM users.

1.3. Organization of the research

Figure 1.1 illustrates the overall structure of this report, and the methodology based on which the research is organised into five chapters:

1. *Introduction*. The theoretical background is discussed, research objectives and questions are clarified. It is followed by presenting the research structure and the methodology.

2. *About Cambridge Advanced Modeller*. Drawn on the historical background, the CAM software is introduced with particular attention to its main toolboxes. For each modelling toolbox, an overview of the relevant literature is presented to demonstrate the path that the software has been passed to be presented at its current version. A brief description of the additional research toolboxes (which are not included in the public version) is also

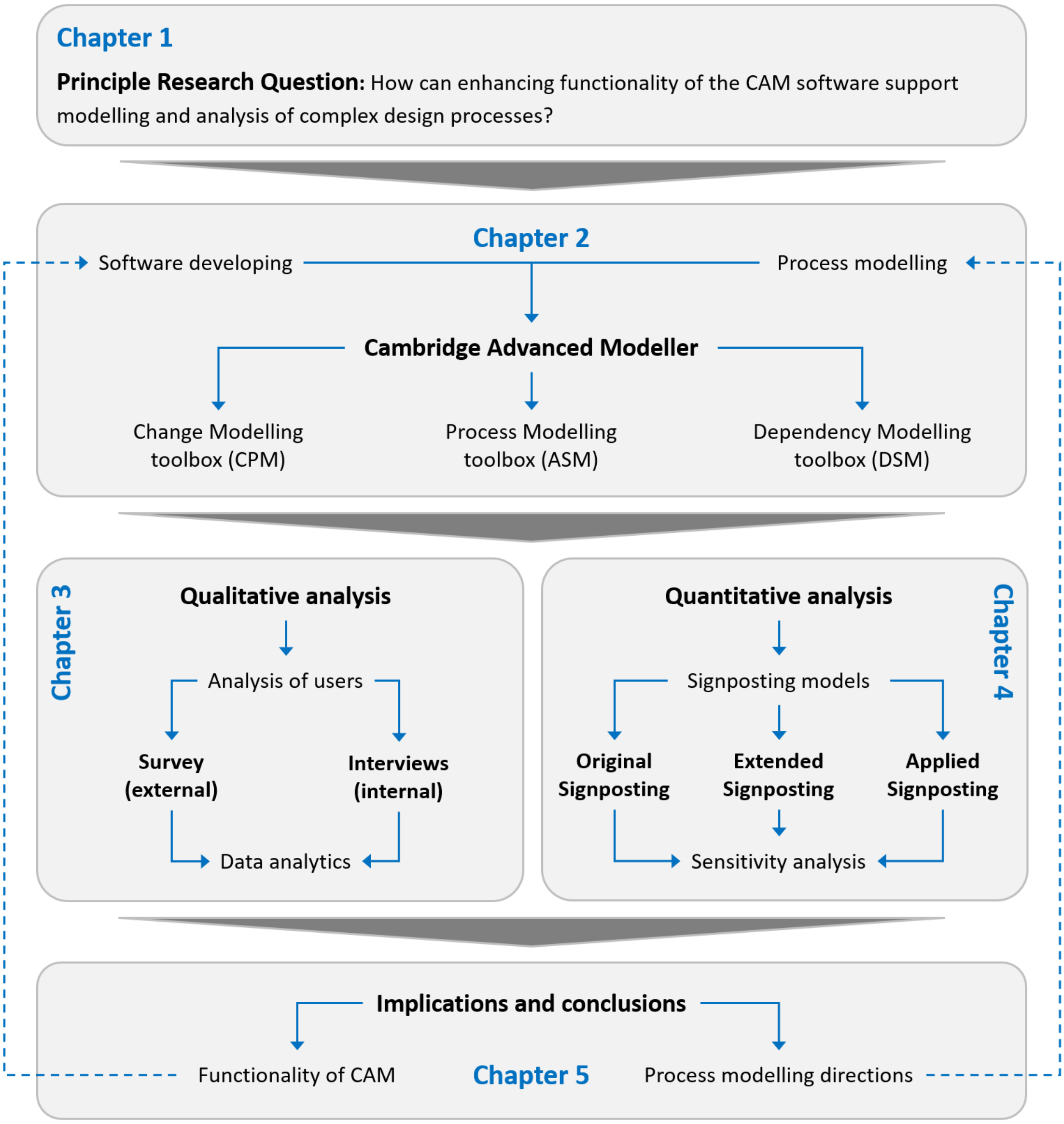


Figure 1.1. Organization of the research: presenting the report structure and the methodology

presented. The outcome of this chapter would help better understand the key characteristics of a modelling tool, thus satisfying RQ1. In addition, it provides a baseline for evaluating CAM, qualitatively and quantitatively.

3. *A qualitative approach for evaluating CAM*. The qualitative aspect of our assessment aims to address RQ2 and RQ3 and starts with an analysis of its internal and external users. Those researchers who have directly been concerned with the software listed and interviewed

in separate individual sessions. In addition, among thousands of external users, a list of so-called active users provided and surveyed using a multi-section questionnaire. By integrating the data from individual interviews and survey, the analytics then provided (through applying a sort of statistical analysis methods, such as factor and regression analyses) and discussed with the development team.

4. *A quantitative approach for evaluating CAM*. The more technical analysis of the CAM software is addressed in this chapter, from the developers' (modelling) perspective. In other words, rather than describing the user needs and preferences, this chapter is gone through the modelling and simulation in detail, to answer the last two research questions, RQ4 and RQ5. In doing so, three different versions of Signposting system, including the Original Signposting (Clarkson and Hamilton, 2000), Extended Signposting (O'Donovan, 2004), and Applied Signposting - ASM (Wynn, 2007), selected. These models are basically different in their assumptions and formulations. The case study of a mechanical design (come from the original Signposting model) used to re-build and re-simulate the models in the Process Modelling toolbox (ASM) of the CAM software. In fact, except for the latter ASM model, which was corresponded to a similar (process modelling) toolbox in CAM, the consistency between the language of the model and the modelling notations (template) in the software was not identified before. Further considerations are then made through a range of experimental analyses.

5. *Implications and conclusions*. This chapter summarises the key findings, recaps the research contributions, and concludes this report. It combines the result of analyses that have been obtained in chapters 3 and 4. It is argued that in spite of the diversity of perspectives in a modelling language, people (with a different range of modelling purposes) have some common interests (expectations). Moreover, by comparing the insights obtained from interviews and survey with the result of the comparative simulation, I stylised the observations and highlighted some issues. As long as the progression of this report, these issues have been discussed in the *Process and Change Management* group and considered in future updates of the software. From a modelling perspective, this chapter discusses that some aspects of modelling have a more significant impact on the simulation outcomes and hence should be considered in any modelling attempts.

Chapter 2

2. About Cambridge Advanced Modeller (CAM)

This chapter introduces the CAM software platform, its architecture including toolboxes, and an overview of the underlying research projects that have been accomplished to empower the CAM's functionality. The main objective of this chapter is to achieve a better understanding of the key characteristics (i.e., functions) of the software. The information provided in this chapter will recap in the following chapters to investigate aspects of DP modelling in CAM.

2.1. Historical background

As is shown in Figure 2.1, CAM, originally known as P3 (product, process, people) Signposting, is a *Java-based software platform* that developed and maintained by the Cambridge EDC to facilitate modelling, simulation, and analysis of the dependencies and information flows in such complex systems (Wynn et al., 2009). The typical example of complex systems would be the DP of products or services, or such design organisations, where simulation and analysis could be helpful to identify possible bottlenecks, optimise processes, and identify unnecessary rework.

The software platform is based on a *graphical diagramming interface* which should be familiar to users of standard office productivity suites (Wynn et al., 2009). It is free of charge for research, teaching, and evaluation purposes. Thanks to its *configuration*-based structure, rather than *programming*-based, it ensures modelling framework to remain stable and reusable while allows easy extension or customisation by the user – since only knowledge of the configuration approach is required, and no knowledge of implementation code (Wynn, Wyatt, et al., 2010). The user interface is constructed automatically from this configuration

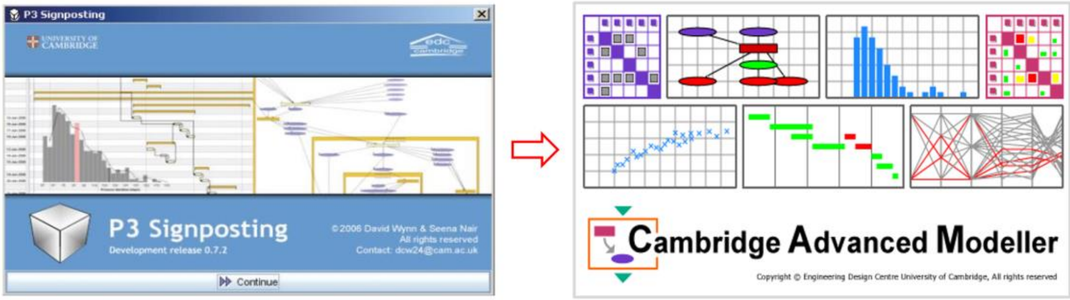


Figure 2.1. The CAM software launching interface (right), originally known as P3 Signposting (left)

to provide an experience tailored to the modelling approach at hand. Therefore, it can be a kind of alternative to those *general-purpose diagramming tools* or *spreadsheets*.

Basically, CAM is a *research-driven* software platform and since its origin, a number of standard modelling approaches and algorithms have been modelled and implemented in CAM, such as System Dynamics, Petri-Nets, and other specific diagramming and matrix-based models. Figure 2.2 shows the screenshots of some of these examples adapted from P3 Signposting. The original P3 platform was a kind of diagramming tool, which was developed based on the concept of Applied Signposting Model – ASM (Wynn et al., 2006), as a graphical tool with underlying mathematical logic. Over the years, as it was evolved, the software allowed modifications to extend its application, through adding toolboxes. Since then, a number of toolboxes have been created, such as the ASM toolbox (adapted from the original P3 platform), a number of DSM toolboxes, and a Change Prediction Method (CPM) toolbox.

From a structural point of view, CAM provides a wide range of functionality, which is organised into *toolboxes*. In particular, its configuration allows to develop new modelling notations and consequently, to add new functions and simulation codes. Each toolbox provides certain features for modelling and analysing certain types of system. For example, DSM toolbox was created to model the dependencies between elements (such as product components, process elements, etc.), or CPM toolbox was developed to analyse the likelihood and impact of change propagation paths (Clarkson et al., 2004), hence supporting change prediction.

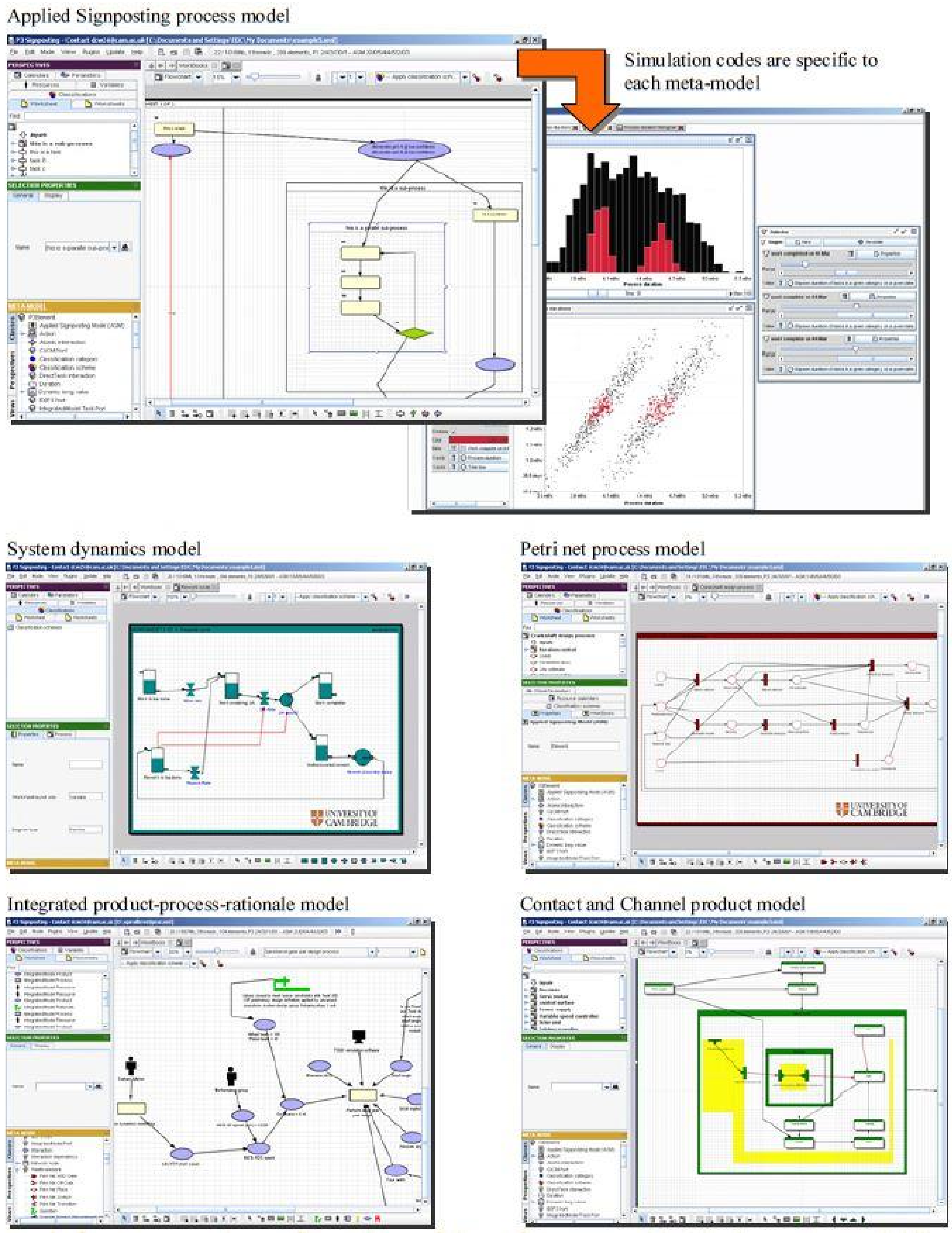


Figure 2.2. Example configurations of the P3 Signposting (Adapted from Wynn et al. 2009)

As of its first public release in 2007 until now, CAM has always been supported by new functions added to its main toolboxes, which are ASM (for *process* modelling), DSM (for *dependency* modelling), and CPM (for *change* modelling). Further information regarding the history of updates and the main functions can be found on the CAM website (http://www-edc.eng.cam.ac.uk/cam/). The website provides detailed information on the way that each toolbox work, the mechanism of downloading the software, as well as other helpful information for researchers and developers.

As mentioned before, the functionality of the CAM software is rooted in the functionality of its toolboxes. Figure 2.3 displays the main toolboxes of CAM. When creating a new workbook, one can select the type of toolbox, by selecting the corresponding model type in the appropriate drop-down list. In the example figure, three workbooks created, respectively ASM (for process modelling), DSM (for dependency modelling), and CPM (for change modelling).

The following subsections present an overview of the research works that have been undertaken over the past years at the Cambridge EDC, in support of CAM or by using it. There are nevertheless some other research toolboxes which are more a kind of plug-in (Java pallet) and due to their particular application, they are not publicly available to the users.

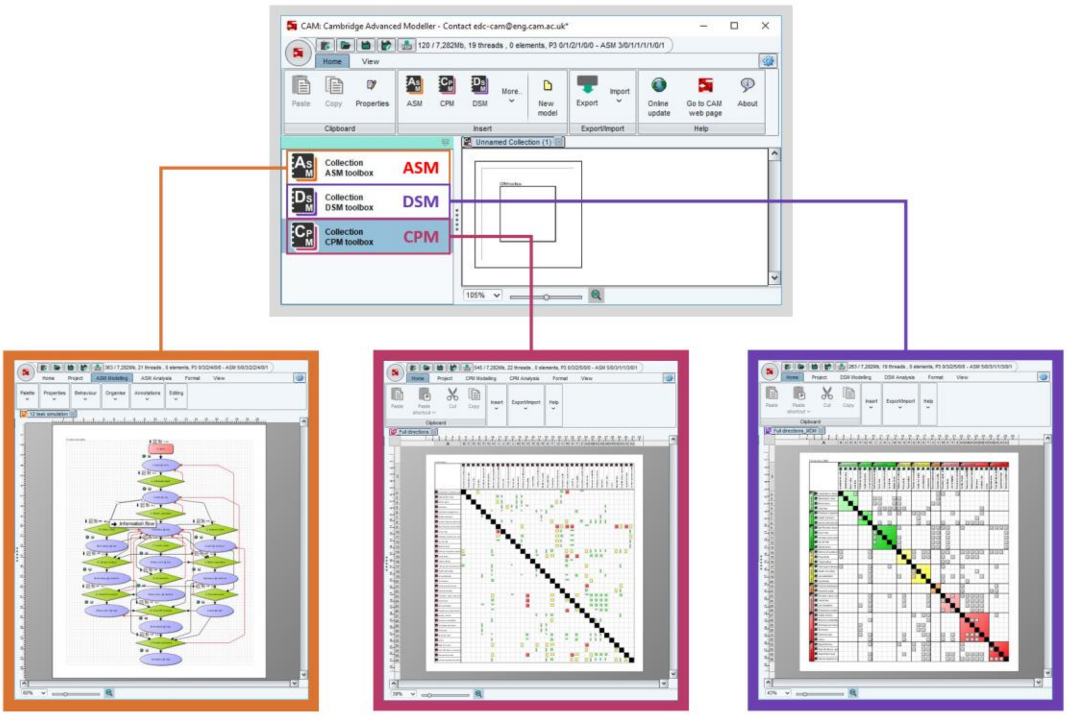


Figure 2.3. Screenshot of the main CAM toolboxes: ASM (left), DSM (middle), and CPM (right); Screenshots are adapted from the research use cases in Cambridge EDC database

2.2. Process modelling toolbox – ASM

The origin of process modelling toolbox refers back to the development of original Signposting approach (Clarkson and Hamilton, 2000). The model was a parameter-based design process model has been developed based on empirical ideas about the use of models in complex design projects. It is termed *Signposting*, since it identifies appropriate routes for the design process, and described along with its implementation and preliminary evaluation.

2.2.1. Overview of the literature of ASM toolbox

Since then, and up to date, advancement of Signposting system has been the focus of several researchers in Cambridge EDC, in a range of collaborative projects with industry partners mostly from Aerospace sectors. A simplified list of contributions associated with Signposting model is presented in Table 2.1. Further to that, the evolution of Signposting models according to their functionality and structure is extensively presented in Chapter 4.1.

The primary extensions of Signposting were mainly concerned with optimisation, identification, and visualisation of process *routes* (Clarkson, Melo, et al., 2001), and expanding the concept of *Confidence* in Signposting (O'Donovan et al., 2003). In fact, these models acted as a baseline for developing the Applied Signposting Model - ASM (Wynn et al., 2006). ASM is a tool to support process improvement, based on a simple graphical notation reminiscent of a flowchart, which was designed to be easy to read for large models and by unfamiliar users (Wynn et al., 2006). It combines the graphical simplicity with the ability to create highly configurable workflow simulation models. ASM provides a diagramme, a simulation tool, and a DSM tool. From this point of view, it can be classified as an integrated tool between the boundaries of Signposting and DSMs.

The outcome of ASM was a software platform developed to implement the ASM, which was originally named P3-Signposting (Wynn et al., 2009). Further research after introducing the ASM attempted to enhance the utility of the model, for example, through expanding the scope of the model and incorporating incorporate life-cycle engineering considerations into design (Kerley et al., 2011), or investigating the effects of management levers on lead time and design errors by integrating ASM with System Dynamics (Le et al., 2012).

Table 2.1. Overview of the research in Signposting, and their relation to the ASM toolbox

Reference	Title	Contribution to the ASM toolbox
(Clarkson and Hamilton, 2000)	'Signposting', A Parameter-driven Task-based Model of the Design Process	The original model of design founded on the assumption that a design process can be constructed from a predefined set of tasks
(Clarkson et al., 2001)	Visualization techniques to assist design process planning	Improving the original Signposting to support planning of natural process constraints, such as bottlenecks, and alternate process routes
(O'Donovan et al., 2003)	Signposting: modelling uncertainty in design processes	Improving the original Signposting through expanding resource to human and non-human types, and widening the concept of Confidence
(Wynn et al., 2006)	Applied Signposting: A modeling framework to support design process improvement	Developing the original ASM model: a tool to support process improvement through description, simulation, and automation
(Wynn et al., 2009)	The P3 platform: An approach and software system for developing diagrammatic model-based methods in design research	Developing a configurable software platform for ASM, so-called P3 Signposting
(Wynn, 2010)	Modelling and simulating a product development process using the Applied Signposting Model in the CAM	An introductory material for getting started in CAM
(Kerley et al., 2011)	Redesigning the design process through interactive simulation:	Developing an iterative simulation environment based on the refinement of ASM according to the updated Gant chart
(Wynn et al., 2011)	Modelling the evolution of uncertainty levels during design	Expanding the cooncept of ASM simulation to multiple levels of uncertainty in an evolutionary process
(Le et al., 2012)	Impacts of concurrency, iteration, design review, and problem complexity on design project lead time and error generation -	An analysis of the effects of management levers on the lead time and design errors by integrating ASM with System Dynamics
(Chen et al., 2016)	Improving Design Resource Management Using Bayesian Network Embedded in Task Network Method	Developing an approach, combining ASM with Bayesian theory, to evaluate sensitivity of resource allocation in design tasks
(Shapiro et al., 2016)	DPCM: a method for modelling and analysing design process changes based on the Applied Signposting Model	Developing a change management method through expanding the concept of confidence mapping in ASM

More recently, Chen et al. (2016) presented a method to model different resource types (designers, computational, testing), by combining the ASM model (as a task-based network) with the Bayesian network, and studied the impact of using different options of those resources. Simultaneously, Shapiro et al. (2016) expanded the concept of Signposting confidence mapping in ASM and developed a change management method to enhance the understanding of DP change effects on process performance. The objective was to support process execution through suggesting mitigating reactions to the changes, and support process planning through identifying and prioritising the *right* changes.

At its current state, ASM is being used for modelling and analysis of the DP of the product portfolio in a collaborative project between EDC and Laing O'Rourke and at the same time, for real-time mapping and mining of data in a collaborative project between EDC and Jaguar Land Rover. In the former project, the goal is to expand the concept of resource management and project scheduling to multi-product (considering a group of people working on parallel projects simultaneously). In the latter case, the mathematical inference behind ASM is being incorporated by the Genetic Algorithm to capture the evolution of confidence in the project parameters over time.

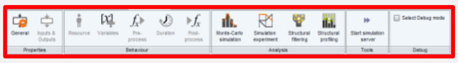
2.2.2. The functionality of the ASM toolbox

As of its original release, many features and functions have been incrementally added to the different toolboxes of CAM including ASM. In the current version of the software, there are two sets of icons in the ASM toolbox representing the modelling and analysis functions. They are listed in Table 2.2, demonstrating the labels, the graphical representation (icon in the software), and a brief description.

The modelling functions are mainly concerned with creating/customising multiple types of tasks and associated deliverables, the interdependencies between them, and the underlying properties such as resources, parameters, and variables. There are three types of tasks allowed to model in ASM: simple, compound, and iteration task (Wynn, Wyatt, et al., 2010). The analysis panel in the ASM toolbox is concerned with running experiments (based on the Monte-Carlo simulation) and designing new experiments.

There is nevertheless a range of materials and articles educating how to get into modelling and analysis of ASM. Examples are Wynn et al. (2009), Wynn et al. (2010), and Wynn (2010).

Table 2.2. The main functions in the ASM toolbox, along with their short description

Label in CAM	Graphical icon	Description of functionality
Modelling icons		
Simple task		... represents tasks which take account of inputs to create outputs. All outputs of a simple task are created concurrently
Compound task		... can have one or more output scenarios or forward branches. Each scenario contains one or more deliverables
Iteration task		... similar to a compound task, but represents the possibility of generating a 'backward branch
Deliverable		... represents packages of information or materials that are considered, created or modified by tasks
Milestone		... represents the gateway, finishing point of a design phase
Connect		... represents the simple connection between tasks, the dependency contributes to the downstream task
Connection to edge		... represents the connection between dependencies
Hyperlink mode		... allows to connect the task to a different worksheet
General		... represents the general properties of the task, such as the name
Input and outputs		... represents the input and output conections of a tasks
Resources		... represents the individuals, teams or other resources that are needed to perform tasks
Variables	[X]	... is used in simulation models to represent KPIs and their interrelationships with tasks in the process
Pre-process		... is a kind of function that allows the user to assign the values to the variables before the task execution
Post-process		... is a kind of function that allows the user to assign the values to the variables after the task execution
Duration		... represents the duration of the task, in terms of a numerical value or a stochastic function
Analysis icons		
Monte-Carlo simulation		... allows to run the discrete-event simulation
Simulation experiment		... allows to setup and run the simulation experiments

2.3. Dependency modelling toolbox – DSM

DSM is a widely used technique across many areas of research and practice that was originally developed with the purpose of modelling and analysing the interdependencies of complex engineering projects such as a product, process or organisation (Smith and Eppinger, 1997). It provides a simple, compact, and visual representation of a complex system that supports innovative solutions to decomposition and integration problems (Browning, 2001).

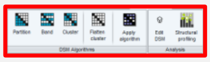
DSM can be a useful tool for discovering and highlighting aspects of dependency structure, such as groups of elements that are strongly coupled to each other. It is also useful for visualising the dependency structure in a compact form. More recently, a long track of DSM usage has led to the development of DMMs and MDMs that have broadened the capabilities and applications of matrix-based models of complex systems and provided further insights. A comprehensive and up-to-date review of the DSM literature can be found in Browning (2016), in which the author provided a high-level review of more than 1000 research items, aiming to consolidate the broad knowledge in creating, manipulating and applying DSMs.

In spite of a large body of research in DSMs, and according to the www.dsmweb.org, majority of the DSM tools in the community are commercial. Amongst those research tools, it has always been challenging to support the vast users with a user-friendly platform that can create, modify, optimise (through partitioning, clustering, etc.), and analyse large complex problems in a reliable and robust way. Due to these reasons along with several motivations for internally using DSMs, a separate DSM toolbox has developed in CAM.

2.3.1. The functionality of the DSM toolbox

The DSM toolbox of CAM is a versatile platform, in the sense that can provide multiple views on a problem. It can be used in CAM in two different ways: as a tool to create a model from scratch or as an additional view of a model which is created in another toolbox. In fact, it has been the most frequently used toolboxes of CAM over the past decade, according to the EDC database, perhaps because of offering advanced modelling and analysis functions free of charge. Structurally, similar to the ASM toolbox, it is organised in the CAM software in two different but interrelated panels: modelling and analysis. They are listed in Table 2.3 supported by a graphical representation and a brief description.

Table 2.3. The main functions in the DSM toolbox, along with their short description

Label in CAM	Graphical icon	Description of functionality
Modelling icons		
Select / move		… represents the selection or moving a DSM element within the same matrix
Submodel		… allows to create a hiearchical DSM
DSM element		… represents a DSM elements
DSM delete		… allows the user to delete a DSM element
DSM connect		… allows the user to create a dependency between two DSM elements
Analysis icons		
Partition		… uses loop searching algorithm to find an ordering such that as marks as possible are below the leading diagonal
Band		… groups subsequent elements in a cluster if they can be attempted concurrently
Cluster		… automatically groups the matrix (or cluster) into strongly-connected sub-clusters
Flatten cluster		… allows the user to remove the clusters from DSM
Apply algorithm		… allows the user to run any of the above DSM algorithms, in a separate window
Order		… allows the user to order the DSM matrix based on a property
DSM structural profiling		… modifies the structure of a CAM model, e.g., it may find and remove all nodes having a certain type

The modelling panel is concerned with creating a new matrix, adding or removing elements, adding or removing layers (to create MDM-like matrix), and making interdependencies between the elements using the *Connect* icon. The analysis icon is mostly responsible for optimising the matrix, e.g., partitioning (based on the *loop-searching algorithm*), clustering, banding, and ordering. For example, the structural filtering functionality uses a graph grammar/set-based approach to modify the structure of a CAM model. For instance, it may find and remove all nodes having a certain type. This allows different perspectives to be generated on the data (further information can be found on the main CAM website, dependency modelling toolbox).

2.4. Change modelling toolbox – CPM

Change is one of the most powerful driving forces in design: a crucial aspect of reaching and maintaining product competitiveness (Eckert et al., 2001). The key challenge during this process is that a change in one part of the system can affect the other parts, sometimes without them necessarily being aware of the source. This would be more challenging when considering technical products are composed of many interconnected parts that work together in geographically distributed way. As the result, changes may require yet more components to be changed. The propagation of change between related elements in a system can nevertheless be observed in other types of system, such as products, processes, and organisations. The levels of impact are shown in Figure 2.4, adapted from Eger et al. (2003).

2.4.1. Overview of the literature of CPM toolbox

The original idea of the change prediction method – CPM – has been presented in the work of Clarkson, Simons, et al. (2001), which was later superseded as a journal paper (Clarkson et al., 2004). CPM is a matrix-based numerical approach for predicting and analysing how changes are likely to propagate through a system. A CPM matrix is composed of the main components of a system and the dependencies between them in terms of likelihood and impact. Figure 2.5 shows an overview of the CPM model.

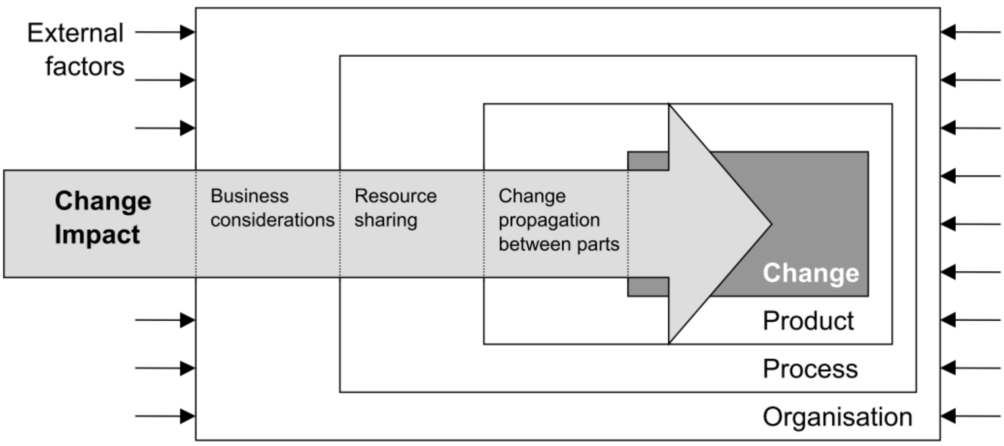


Figure 2.4. The levels of change impacts in a complex system (adapted from Eger et al., 2003)

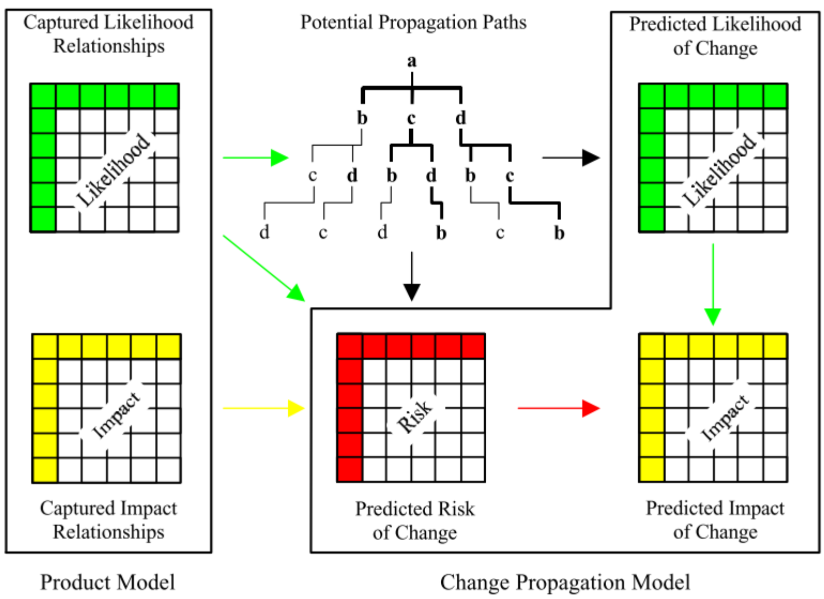


Figure 2.5. Overview of the CPM model (adapted from Clarkson et al., 2001)

In fact, likelihood and impact values inside the matrix-cells represent the strength of the linkages regarding the propagation of engineering changes (Clarkson, Simons, et al., 2001). Central to the CPM approach is a stochastic algorithm – the Union Operator algorithm – that applies to calculate the combined risk of change propagation between components considering multiple steps of direct and indirect change propagation (Figure 2.5). Based on this product model and the combined risk values, CPM generates different diagrams (such as distance network, propagation path, risk plot, risk network, and risk portfolio) to support the change propagation analysis.

Before developing the CAM, the CPM methodology was performed on a software prototype. By developing the ASM methodology and its associated software (P3 Signposting), the two methodologies, together with the DSM toolbox, they all then incorporated in an integrated software platform: in CAM. Nevertheless, the original CPM was concerned with many assumptions which could be seen as limitation towards the validity of the model. Bringing the full validity of the model and further improvements to the original CPM have been the focus of many studies the Cambridge EDC afterwards. Table 2.4 represents some of the key studies in this area.

Table 2.4. Overview of the research in Change Modelling and their relation to CPM toolbox

Reference	Title	Contribution to the CPM toolbox
(Eckert et al., 2004)	Change and customisation in complex engineering domains	A qualitative investigation of change propagation in helicopter design leading to the development of CPM method
(Clarkson et al., 2004)	Predicting change Propagation in complex design	A quantitative investigation of change propagation leading to CPM method; for predicting change paths in complex systems
(Keller et al., 2009)	Using an engineering change methodology to support conceptual design	Development of a CPM model to support changes in the diesel engine conceptual design stage
(Hamraz et al., 2012)	A multidomain engineering change propagation model to support uncertainty reduction and risk management in design	Proposal for a multidomain model which combines concepts of both the function-behavior-structure (FBS) model with the CPM
(Koh et al., 2013)	A technique to assess the changeability of complex engineering systems	Development of a matrix-based approach to generate change indices for individual components of a system
(Ahmed et al., 2013)	Change impact on a product and its redesign process: A tool for knowledge capture and reuse	Developing a knowledge management to reflect organisational dynamics, resulted to a new research pallet in CAM
(Hamraz et al., 2015)	FBS Linkage ontology and technique to support engineering change management	Development of an ontology for function, behaviour and structure (FBS) linkages and its application to the case of a diesel engine
(Hassannezhad et al., 2017)	Dynamic modelling of relationships in complex service design systems	Proposal of a systematic approach integrating the functionality of CPM with Systems Dynamics to cope with complexity issues

The focus of some of the studies such as the work carried out by Keller et al. (2009) and Koh et al. (2013) was to investigate changeability of multiple design contexts to support the identification of criticalities, i.e., key elements and key dependencies. Some others such as the works of Hamraz et al. (2012) and Ahmad et al. (2013) concentrated on expanding the scope of CPM to multiple domains. For example, in the research of Hamraz et al. (2012), the authors presented a multi-domain CPM by combining the concepts of CPM with the well-known *Function-Behaviour-Structure* (*FBS*) ontology to control change propagation and reduce uncertainty and risk in design. Eventually, their work resulted in the development of an ontology for FBS-CPM linkages (Hamraz et al., 2015).

Very recently, Hassannezhad et al. (2017) combined the concepts of CPM (a matrix-based method) with Systems Dynamics (a network-based method), as a way to cope with complexity of large mature organisations through focusing on the key system elements (in terms of their impact in propagating changes as the result of applying CPM) and understanding their dynamic behaviour, i.e., how these key elements can affect each other and other elements in the network. In their work, the authors developed a multi-domain CPM regarding the strategic, tactical, and operational levels of decision-making in the organisation, and from the perspectives of the organisation, employee, and customer.

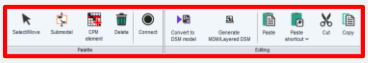
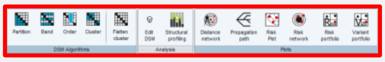
Currently, the CPM toolbox is being used at the Cambridge EDC in several industrial projects, for instance, for understanding the nature of changes in designing resilient manufacturing design (a collaborative project with a major construction company), or for reasoning the underlying risk of decisions in a complex network (a collaborative project with a major telecom company).

2.4.2. The functionality of the CPM toolbox

There is a close relationship between the DSM and CPM toolboxes of CAM. As mentioned before, DSM can be created in the CPM toolbox as an additional view of the problem. At the top of that, a CPM model can be constructed based on an existing DSM, as CPM is essentially a kind of DSM-based methodology. Therefore, the modelling elements of the two toolboxes are similar so far. The full list of modelling and analysis icons of the CPM toolbox is shown in Table 2.5, supported by a graphical representation (icon) and a brief description.

In particular, the modelling icons in CPM are responsible for creating the CPM matrix through adding the elements (i.e., system components), connecting the elements, and adding the impact and likelihood associated with each connection node. The CPM algorithm applies the stochastic algorithm to calculate the combined risk of change propagation between elements (Clarkson et al., 2004). Further configuration or customisation of the model can be accomplished by using the post-process simulation icons (Table 2.5, analysis icons). The outcome of the model is a range of diagrams and matrices that can be used both for re-architecting the problem as well as for further sensitivity analyses.

Table 2.5. The main functions in the CPM toolbox, along with their short description

Label in CAM	*Graphical icon*	*Description of functionality*
Modelling icons		
CPM element		… represents the creation of a CPM element
CPM delete		… allows the user to delete an existing CPM element
CPM connect		… allows the user to make dependency between two CPM elements
Analysis icons		
CPM algorithm		… allows the user to run the CPM algorithm
CPM experiment upload		… allows the user to run additional experiments on an existing CPM
CPM structural profiling		… modifies the structure of a CAM model, e.g., it may find and remove all nodes having a certain type
Distance network		… represents the output of CPM algorithm, in terms of visualising the distance between any two elements
Propagation path		… represents the output of CPM algorithm, in terms of visualising the propagation path between two elements
Risk plot		… represents the output of CPM algorithm, in terms of visualising the compound risk plot
Risk network		… represents the output of CPM algorithm, in terms of visualising the risk network
Risk portfolio		… represents the output of CPM algorithm, in terms of visualising the risk portfolio
Variant portfolio		… represents the output of CPM algorithm, in terms of visualising the variant risk portfolio

2.5. Further research toolboxes

The previous sections presented the key toolboxes of the CAM software. However, some other toolboxes have been introduced in CAM that can be termed as plugins. The examples are the Exploring Possible Architectures (EPAs) toolbox (Wyatt et al., 2012), the Decision Rationale editor (DRed) toolbox (Aurisicchio and Bracewell, 2013), Topic Maps toolbox (Stevens, 2012), Organisational Dynamics toolbox (Wynn et al., 2012), and a simulation model to evaluate the benefits of change prediction for scheduling (Wynn, Caldwell, et al., 2010).

These plugins are the outcome of studies at the Cambridge EDC that are not publicly available for use. Because they might have been the outcome of a collaborative project with a particular company and there is a kind of confidentiality concerns for their public release (e.g., DRed), or validation of these plugins have not been approved by the group in several case studies and hence, requires further calibrations for public use (such as Organisational Dynamics and change prediction for scheduling). More detailed discussing of these toolboxes is out of the scope of this report. So they will not be further discussed in the following chapters of this report.

2.6. The context at the Cambridge EDC

This chapter was about to introduce the CAM software package, its main (and the associated) toolboxes, and the research contributions undertaken to provide the toolboxes as it is available nowadays. In fact, this work is officially the first ever contribution addressing aspects of CAM in detail.

It should be mentioned that research in aspects of CAM has been mostly the focus of the Process Management and the Change Management groups (amongst 13 different research groups, each one addressing a particular aspect of design) at the Cambridge EDC (further information regarding the centre and its research groups is available in the main EDC website). It is also the key research tool that can be used for multiple research purposes. In general terms, these groups research the role and nature of the process and change modelling in the successful delivery of new products and services.

During this chapter, it is mentioned that the CAM toolboxes have been developed and improved in EDC over the years based on a close collaboration of research between the centre and multiple industry partners, ranging from aerospace and automotive sectors to construction, consumer electronics, and telecommunication sectors. Attempts made to provide a reliable solution for solving real-world problems, especially more complex ones.

The following chapter presents the structure and outcome of the qualitative analysis of the software to understand what is more important for thousands of worldwide users. Accordingly, Chapter 4 will present a more detailed and quantitative analysis of the software based on a real case study.

Chapter 3

3. A qualitative approach for evaluating CAM

The previous chapter presented an overview of the research in process and change modelling at the Cambridge EDC that led to the development of the CAM software. Accordingly, the main toolboxes of the software explained and supported by the demonstration of the functionality of each toolbox. Built on the previous discussion, the presenting and the following chapters look at the functionality of the software in more detail, and respectively from a qualitative and a quantitative perspective.

Doing this way, this chapter focuses on the role of software users (i.e., their expectations and feedback) and aims to understand (1) the key characteristics of the software (RQ.2), (2) understand those aspects of the software that have been more popular between external users (RQ.3), and (3) provide some insights for future improvements of the software based on the analysis of feedbacks (RQ.4). Explicit understanding of these purposes can provide the users with a better understanding of the full functionality of the software and eventually, help broadening the utility of the software. It is also helpful to explore its potential capabilities through understanding the major improvement points for its future releases.

To address these issues, the rest of this chapter presents the development and application of a framework for the qualitative analysis of the CAM software, including the analysis of data (ranging from early 2010 to the end of September 2016), observations from the survey, and the discussion on findings.

3.1. An empirical framework for evaluating CAM

Figure 3.1 outlines the proposed qualitative approach. In terms of the methodology, a differentiation firstly made between internal and external users of the CAM software. The internal users were those people who are using or have been using the software at the

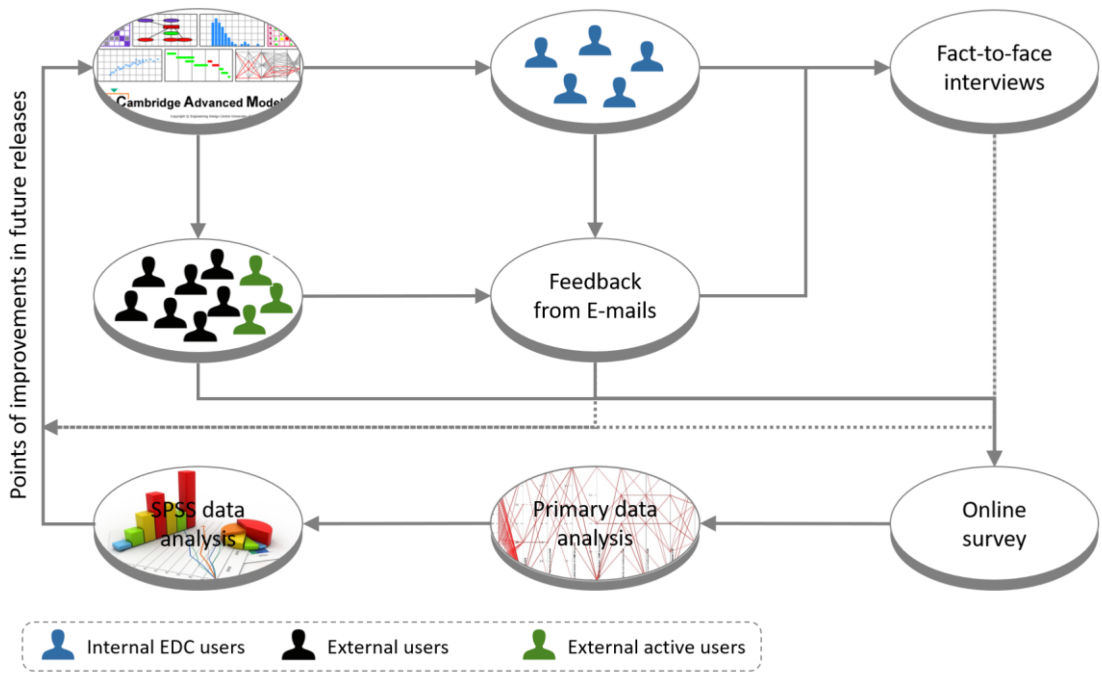


Figure 3.1. The proposed approach for the qualitative evaluation of CAM

Cambridge EDC in the past few years. The external users were people who are using or have been used it outside of the EDC since early 2010. After that, referring to the CAM database, the previous E-mails (from external users) received by the CAM support team collected and reviewed to find out the most important concerns that have been reported by users in the past – typically the internal/external users contact the CAM support team for any query or problem facing the usability of CAM. This data then used to design and conduct a range of interviews with the internal users whom are/have been concerned with any aspect of CAM and its toolboxes currently/at any time in the past.

In parallel to this phase, I analysed the users' data according to the frequency and date of downloads, and the affiliation of users who download the software for the period of early 2010 to the end of September 2016. The following Chapter 3.2 presents the outcome of primary user analysis. This data along with data collected from the interviews (Chapter 3.3) then used to design and release the online survey (Chapter 3.4).

For the purpose of CAM analysis, there have been three types of data that gathered and used in this study come from: the users' E-mails (queries), the face-to-face interviews with EDC colleagues, and the online survey. Nevertheless, all this data transferred to the CAM support team at the end of this study to support them in future improvements of the software. Eventually, the result of the online survey utilised for further statistical analysis such as factor analysis and parallel coordinates visualisation and will be presented later in Chapter 3.5.

3.2. Analysing the composition of CAM users

All the information of the CAM users obtained from the IT team. It was a long list of raw data that has to be organised and meaningfully reconfigured. After retrieving the information in Excel, the analysis team (specifically the author supported by an advisor) first filtered the data for the period of early 2010 to the end of September 2016 and removed the rest including the useless data such as the incomplete or unspecified user profiles. Then, the information sorted out according to the name, affiliation, download date, and users' E-mail address. At this step, I got a number of 5495 total downloads, with 3498 unique users amongst which. The composition of users is illustrated in Figure 3.2 (a-d).

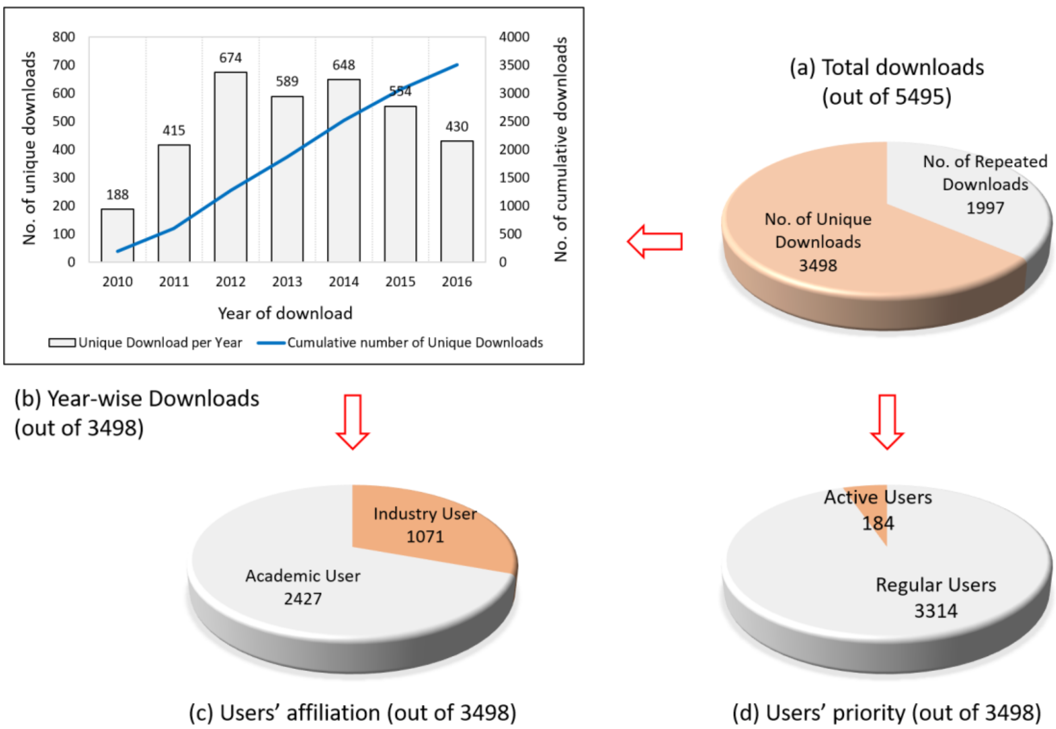


Figure 3.2. The composition of CAM users: (a) Total downloads, (b) Year-wise downloads, (c) Users' affiliation, and (d) Users' priority, according to their number of downloads

The first classification applied based on the uniqueness of downloads, since I found many users with several downloads of the software in different years, e.g., downloading the newest version of CAM every year, or downloading on different versions of operations systems. By merging the downloads per person, I got 3498 unique downloads as the result (Fig. 3.2-a). Its distribution over the years of study is demonstrated in Fig. 3.2-b, in which the period of 2012-2014 shows the highest number of downloads. However, investigating the variation of CAM users over the years is out of the scope of this research.

Another classification made by distinguishing the affiliation of users at the time of using CAM. It was interesting to the support team to see that almost one-third of the downloads made by users with an industry affiliation (Fig. 3.2-c). The final classification was actually related to finding a master list of users for the online survey. Therefore, I considered the number of downloads per user and then prioritised them in the following sub-classes:

- Active users with *high* priority: whom with at least five downloads in consequent years;
- Active users with *medium* priority: whom with three to four downloads in consequent years;
- Active users with *low* priority: whom with less than three downloads in consequent years;
- Regular users: whom with single/multiple downloads in the same year;

The belief was that these users have actively been using CAM on a project/research for a period of time which is long enough to provide the user with a deep understanding of the functionality of the software. As the result, 184 users (out of 3498) recognised as active (Fig. 3.2-d), 48 number of which with high priority, 89 number of which with medium priority, and 47 number of which with low priority.

The analysis team considered the fact that not all the users who downloaded the software might have not been used it in practice. Understanding the number of actual users of CAM is a difficult task; however, those users who downloaded CAM for at least three consequent years are very likely to be the actual user of the software. These people recognised as the target community of online survey afterwards.

In addition, and to provide a more complete list of users for online survey, the analysis team concentrated on the 430 users that download CAM during the last year of study – first half of 2016 (last column in Fig. 3.2-b). Their information also collected for the online survey.

Overall, analysis of users at this step gave us a master list of 614 users, 33 of which were duplicates and placed both in the list of active users and in the list of 2016 downloaders. They removed from the list and the information of the rest (essentially their E-mail addresses) collected to be used in the online survey (Chapter 3.4).

3.3. Conducting a range of interviews

The advantages of interviews in getting a deeper understanding (e.g., various perspectives) on a subject is well understood in different disciplines. For instance, they allow respondents to reflect and reason on a variety of subjects. For interviewers, it has the potential merit of having direct control over the flow of (primary) data collection and having a chance to clarify certain issues during the interview process.

However, finding the right interviewees and scheduling the meetings might be very time- and effort-intensive, and in some case like ours, completing the process would be out of time scope of the research. In the context of this research, the analysis team decided to focus only on the internal EDC users of CAM for a range of *detailed* interviews, mainly because of their availability and more familiarity with CAM. I then used the outcomes of interviews to design the online survey.

In doing so, a list of 10 people in the Process and Change Management groups of Cambridge EDC provided including four PhD candidates, three early-career researchers (Research Associates), and three senior researchers (Senior Research Associates). All of them have been completely familiar with different aspects of CAM and its main toolboxes.

The methodology that I used in interviews was in fact a kind of looking-backwards looking-forward procedure, in the sense that in addition to the general questions on the functionality of the CAM toolboxes, the analysis team used the contents of the E-mails (queries) that were received by the support team – it is unpublishable, due to the confidentiality concerns – to discuss the limitations and potential improvements of the software in future updates.

Concerning the interview process, a set of *one-to-one* and *face-to-face* interviews with the duration of minimum 30 minutes up to one hour arranged with the interviewees. During the interview, I discussed different aspects of working with CAM, such as its interface, functionality, and from the perspectives of model building, model simulation, and visualization and making/exporting a report. It should be mentioned that attempts made to start interviews with the PhD candidates who were rather less experienced in using CAM toolboxes. The idea was to pursue the interviews in an incremental manner in such a way that the feedback from more junior users used to discuss further points of software improvement with more senior users.

After collecting all the feedbacks, the data refined for analysis. The information then prepared to elaborate the general concept of designing survey, through combining separate descriptions to formulate a coherent narrative. The summary of interviews is presented in Table 3.1.

According to the table, and in terms of model building, the EDC users encountered some issues pertaining to the customisation of toolboxes to address a specific problem, the transferability of information to be used in a different toolbox, and the mathematical engine in the ASM toolbox. Concerning the simulation, exchanging information between different toolboxes is again an issue. This might be due to fundamental configuration of CAM software in that different toolboxes are introduced as different pallets and hence, there is no direct link between the outcome of different toolboxes so that can be used in another toolbox. Optimising DSM (especially for large-size problems) is another issue. However, to the knowledge of authors of this report, a large amount of effort has put at the time of

Table 3.1. The highlights of interviews with internal EDC users of CAM software

Perspective	Comments from the interviews
Model building	Tutorial and training material (mainly for ASM and CPM toolboxes) Functionality during the modelling (e.g., inserting new functions in ASM) Extending CAM to generate an MDM-based CPM directly Creating a pallet of Object Process Methodology (OPM) for CAM
Model simulation and analysis	Interface and tracking mechanism (especially from ASM to DSM) Changing views between different toolboxes Partitioning, banding, and clustering in DSM
Model visualization and making report	Compatibility and integration with other software packages Export issues (e.g., from DSM out onto PDF) Import issues (e.g., import file from excel to CAM)
General concerns	Installation of the software in different platforms Commercial usage conditions The new version is much more user-friendly Availability of further plugins and extensions for research

writing up this report in EDC to address this issue. In terms of visualisation of outcomes, in fact, there has been a significant improvement in providing multiple and customisable views of the outcomes, each one of which representing a specific set of information.

3.4. Developing an online survey

The previous sections explained the process that was passed to prepare the online survey. The feedbacks from (internal and external) users first collected through reviewing the E-mails. This information then used, together with the prior knowledge of the analysis team, to design the interviews with internal EDC users. After that, all this information retrieved and analysed to design an online survey to get feedback from the external CAM users. In particular, the analysis entailed classifying, comparing, and combining the original data in order to reveal patterns into a coherent narrative.

The sample of questionnaire is presented in Appendix. The Google Forms platform used by the analysis team to design and distribute the online survey. The survey distributed around mid-January 2017 with a two-month notice to get the response. The sample of the survey can be found at the following link:
https://docs.google.com/forms/d/14zlNQgoulzdTnOUWHm1fv8iS6n6W8EEgxheJ_YnhwMI/.

3.4.1. Structuring the survey

In terms of the structure, as is presented in Appendix, the survey is designed into three consequent sections, and submission of the survey is subject to filling up all the fields. It starts with *five* questions about personal information. They are mainly related to the gender, age, educational status, and employment status, among which age, educational background and employment status (at the time of using CAM) are mandatory to respond. The goal of the questions in this section is to get an understanding of the general characteristics of the active users and their relevance to the software.

The next section is about the previous experience of the user in using CAM (looking backwards). It contains *six* basic questions, mainly related to the period of use, objective, the version that was used, the operating system, and the overall experience of CAM and its main toolboxes. This section follows by the third section, which is about the user's feedback for getting a further improvement of CAM.

The third section contains *seven* question and in terms of the level of detail, it has gone through the detailed experience of using CAM toolboxes comparing to the previous sections. Much of the questions designed in this section routes in the result of interviews and the feedback from the E-mails. Therefore, they are concerned with the perspectives of model building (including the installation), model simulation, and visualisation and making a report. At the end of the survey, a piece of information about the next update is given to the user and a subscription link is presented to let the user an opportunity of receiving updates and news in future releases of CAM.

Overall, in structuring of the survey, attempts made to consider the coherency between sections (from abstract to rigorous) while keeping the objectives of the survey in mind. It should be designed in a way that satisfies the objectives in terms of getting sufficient information on different aspects of using CAM and at the same time, took not more than five minutes of the users' time to fill out (to get a higher chance of getting a response). As it was expected, several iterations applied to the online form to make it ready for the public release.

3.4.2. Distributing the survey

As mentioned in Chapter 3.1, the survey was distributed in two rounds: in the first round, I focused on the (184) active users and in the second round on the users who downloaded the software in the last year of study (2016). Hence, an E-mail prepared – including the request, potential benefits of responding, and the corresponding link to get into the survey – and sent to 581 users (184 active users plus 430 last-year downloaders minus 33 duplicates in the two lists).

Unfortunately, not all the E-mails successfully sent to the users and, perhaps because of changing the employment status, some of them were inaccessible. For some others, I received an automatic response implying that the user is out of work due to the personal/work reasons. Nevertheless, 533 E-mails sound that successfully sent at the end.

Whatever the reason was, unfortunately, the rate of response was much lower than the expectations, with 53 responses received after the first three months of release (around 10%). From the software support perspective, however, this amount of response was sufficient to understand the main improvement points: what is working very well and what might go wrong? In fact, it was the first ever of its kind to get a feedback from the external CAM users using an online survey. It was nevertheless promising since all the data (come from E-mails, interviews, and survey) brought together for the purpose of this study and eventually for future updates of the software.

The primary statistics of the survey disseminated by Google Forms are presented in Figure 7.1 (as a set of charts and diagrams) in the Appendix. The figure comprises into three parts, each one of which is corresponding to a section in the survey that discussed before. Not all respondents answered all of the questions. Therefore, percentages reported in the charts are related to the total number of users answering an individual question. Overall, the result in Fig. 7.1 implies that the full response to 11 (out of 18) questions is achieved.

For the purpose of this research, a few points from the figure is highlighted in the following bullets, while a more comprehensive (statistical) analysis of the output data is presented in the next section.

- The majority of respondents were from an engineering background (42 out of 53, 79.2%). It is not surprising for a software that is aimed to solve complex engineering problems;
- The same was true for employment status at the time of using CAM, representing that the majority of responses were from the academic institution (27 out of 53, 50.2%). The second largest group were industrial or manufacturing company (18 out of 53, 34%);
- Almost two-thirds of the respondents had less than a year of experience in using CAM (35 out of 53, 66%), and only *six* users found to use it more than three years (12%);
- The same number was true for the main objective of using CAM, and almost two-thirds of respondents were used CAM for study or research purposes (35 out of 53);
- The DSM toolbox has been reported as the most frequent and most popular (in terms of satisfying the users' expectations) toolbox;
- The dominant feedback on the functionality of CAM was related to the data analysis panel in DSM and building and simulating a model in ASM. The CAM support team found that improving the interface and providing a tutorial (especially for more sophisticated toolboxes like ASM) sound very relevant.

Overall, taking both the summary of interviews (Table 3.1) and the survey (Table 7.1) into consideration implies a sense of consistency in all aspects of model building, simulation, and visualisation between the two approaches. Offering a tutorial and video clips have been the preferred way of learning CAM amongst users. Building and simulating a new model as well as the software interface reported as the most important concerns when using the software, particularly when using the ASM toolbox. Amongst the DSM and CPM users, optimisation and analysis of DSM models were of most concern for improvement. In fact, all these issues can be seen in the list of highlights from interviews in Table 3.1. The following section presents a more detailed investigation of the results, mainly regarding the dependencies between responses and the influencing factors (questions).

3.5. Analysing the survey results and discussion

The primary observations presented earlier in this section considered the survey questions independently. This section provides a more detailed investigation of the survey outcome to find out the most influencing questions in terms of their correlation with other questions (through factor analysis) and the visualisation of those influencing factors (as the result of factor analysis) across multiple dimensions (questions) through parallel coordinates diagrams. The objective of this analysis is to give the CAM support team some insights on possible ways that the audience of the software can be broadened and that the functionality of the software can be improved.

3.5.1. Factor analysis using SPSS

In a general context, one of the purposes of factor analysis is to identify underlying variables or factors that explain the pattern of correlations within a set of observed variables. An alternative way to the factor analysis is Principle Component Analysis (PCA).

In principle, firstly proposed by Pearson, the PCA has mostly been used in exploratory data analysis: it is very similar to the traditional factor analysis, but more reliable and conceptually less complex in terms of the structure. PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of correlated variables (questions) into a set of values of linearly uncorrelated variables called *principal components*. This transformation is defined in such a way that the first principal component has the largest possible variance.

More information regarding the PCA method can be found in references such as Field (2009). In this report, as mentioned before, the PCA method is used to find out the most influencing questions in terms of their correlation with the other questions. In the following, I briefly explain the factor analysis process using the SPSS platform and highlight some results.

Data preparation and settings. The survey was contained 18 questions all of which were a kind of qualitative. Rather than multiple-choice questions, some others were further split into sub-questions, for example, educational degree (split into Bachelor, Master, and PhD), rating the main toolboxes (split into ASM, DSM, and CPM). These sub-questions were considered as a separate variable during factor analysis. This was altogether resulted into a set of 23 variables (including both the questions and sub-questions) to be used in SPSS as the inputs. Table 7.2 provides a list of factors along with their description. Doing so, the qualitative data was then converted into the ordinal scale and the factor analysis was applied.

Factor extraction: The total variance matrix presented in Table 7.3 (in the Appendix) shows the outcome of primary PCA, based on the Kaiser's criterion (Field, 2009). The extraction of variables is adjusted to be performed based on the eigenvalue of the component. According to the table, 9 factors is obtained with the eigenvalue of greater than 1.0 which account for around 72% of the total variance. The primary clustering of the variables is represented in Table 7.4 as the Rotated Component Matrix, based on the Varimax method for rotation. Broadly speaking, the two common orthogonal techniques of rotation are Quartimax and Varimax. In this study, the latter Varimax has been used for factor rotation, since it has been reported as the best method to create more interpretable clusters of factors (Field, 2009).

The rotated matrix of principle components (obtained from Table 7.3) shows the clusters of variables (factors) according to the correlation amongst them. In order to extract the least-correlated factors from the analysis, the component(s) that are located out of the clusters (for example, Q5, in Table 7.4) removed from the analysis and the next replication ran. The process continued until getting a sufficiently-correlated cluster of components. Further iterations of the factor extraction are summarised in Table 3.2. The objective of this iterative process was to attain an optimal and simple structure to define a distinct cluster

Table 3.2. The process of factor analysis and extraction until getting the optimal cluster: in each replication, the PCA method applied for the remained factors, and using the same settings as the original analysis presented in Table 7.4.

Replication number	*No. of iterations to converge the rotation*	*No. of clusters (components)*	*Factors that extracted from the analysis*	*No. of remained factors*
First	17	9	Q5	22
Second	23	9	Q1, Q10, Q11, Q23	18
Third	23	8	Q20, Q21	16
Fourth	10	6	Q2	15
Fifth	6	6	Q9, Q16, Q19	12
Sixth	6	5	Q7	11

of interrelated factors so that interpretation be easier, without changing the underlying solution (Field, 2009). Eventually, this enabled us to determine the most influential factors to be used for further investigation in the next section.

Regarding the Table 3.2, after six round of replications, 12 factors totally extracted from the analysis while the number of clusters reduced from 9 down to 5. The final replication of the PCA is presented in Table 7.5. The table shows 5 different clusters of the factors with relatively large amount of correlation between factors within each cluster – the minimum value of correlation in Table 7.5 is equal to 0.647. In the first two clusters, it is interesting to see the functional aspect of the software (in Section 2 of the questionnaire); in particular the overall experience of the users in using CAM toolboxes and their educational background. Next to them, the fourth and fifth clusters of the table are mainly related to the Section 3 of the questionnaire: the improvement points for future releases. Being specific, issues pertaining to the installation and the overall support of the software located in the third cluster; while further improvement points of the software and the duration of using CAM found in the same (forth) cluster. In the last fifth cluster, two factors related to the educational degree of the users are placed.

Overall, the factor analysis presented in this section attempted to provide rather a more detailed understanding of the impact of any of the questions in the survey, based on the correlation between responses. As expected, the more junior users (in terms of the duration of using CAM) had more attention on getting started with CAM through using a tutorial. The more senior users, on the other hand, had particular attention on understanding the full functionality of the software, and the compatibility of the toolboxes with each other and with other external software packages. The following section will get deeper insights into the discussion through parallel visualisation of the results.

3.5.2. Multi-dimensional analysis using Parallel Coordinates

Previously, I applied the factor analysis to determine the most influencing questions of the survey in terms of their correlation with other questions. In this section, I used Parallel coordinates Diagrams (PCD) to translate the responses to the questions into patterns to be used in future improvements of the CAM software.

PCD is a basically visualization technique used to plot individual data elements across many dimensions, typically shown as parallel lines equally spaced. Each of the dimensions (each question in the survey) corresponds to a vertical axis in PCD and each data element is displayed as a series of connected points along the dimensions/axes. As far as related to the functionality of CAM, PCD is one of the very newly added plugins to the software (in order to improve the visualisation part of the software) providing sophisticated patterns of outcomes based on multi-dimensional inputs.

In order to visualise the relationship between responses, I used the insights from the factor analysis to filter the dimensions (questions). By taking the composition of the final cluster of the questions (presented in Table 7.5), a mixture of questions selected as the dimensions in PCD, their corresponding data converted into a CSV file, and then imported into the CAM software. Table 7.6 shows the list of dimensions (questions) whose responses are used for multi-dimensional analysis in PCD. Particular attention made to the questions pertaining to the functionality of CAM and future improvement points.

Nevertheless, the observations can be interpreted in many different ways. In the following, three different interpretations of the observations are presented in Fig. 3.3, as example.

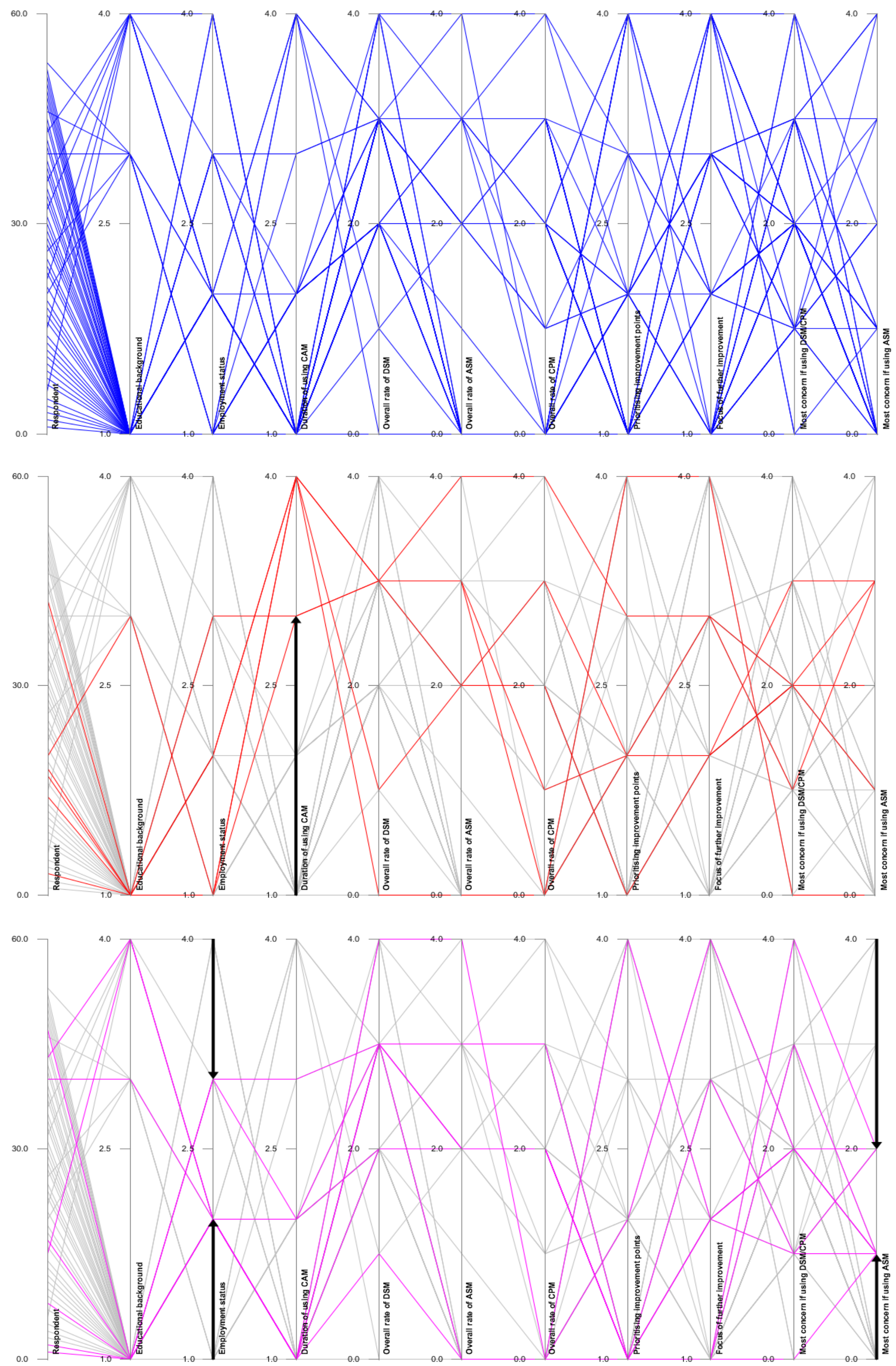


Figure 3.3. Multi-dimension analysis of results using Parallel Coordinates: (top) the complete view, (middle) the restricted view to only the experienced users: those with more than 3 years of experience of using CAM, and (bottom) the restricted view to the ASM users from company whom have been concerned with building and simulating a model.

- The top PCD in the figure displays the complete view of the diagram. A few patterns can be enumerated from the diagram at the first look: the frequency of the users from engineering field is significant; not surprisingly, there is a higher frequency of DSM users comparing to the CPM and ASM users – this is also observable from the last two dimensions; and in terms of improvement points, users with different levels of education and background seem to reflect different considerations on the functionality of CAM. To explore these considerations in more detail, the two other diagrams (Fig. 3.3, middle and bottom) provide a more restricted view of the observations.
- The middle PCD in Fig. 3.3 shows the restricted view of the users with more than 3 *years of experience* of using CAM toolboxes. Its frequency is not high (6 out of 53), but it is worth looking at how more knowledgeable persons criticise the software. These users represented the *interface* and *performance* of the software as their priority points for improvements; in particular, data analysis (e.g., clustering and partitioning in DSM) and generating outputs. This is consistent with the output of interviews with internal users (Table 3.1). It is interesting that unlike the less-experienced users of CAM (users with less than a year of using CAM toolboxes), the experienced users had less concerns with the interface and building of a model. in addition, looking at the dataset shows that the more experienced users of CAM all of them have been used the software for research purposes.
- The bottom PCD in Fig. 3.3 represents the restricted view of the CAM users with an industry affiliation whom have been concerned with building and simulating a model in the ASM toolbox. This view shows that, except one respondent, all of them had less than 3 years of the experience of using CAM. Moreover, all of them have been using the DSM toolbox together with the ASM toolbox – while being satisfied – and considered the compatibility and integration of CAM with other software packages as well as data analysis as their most important concerns. However, most of them counted the interface and ease of use as the priority points for further improvements.

By the end of this phase of the study, similar patterns discussed and delivered to the CAM support team. At the time writing this report, the support team has had a particular attention to the interface of the software, its compatibility with other software packages (such as MS Excel), and the functionality of the software.

3.6. Summary of the chapter

Developing and maintaining a software that is specialised in dealing with complex design problems is a very valuable and of course challenging task. Accordingly, analysing the full functionality of such tools requires a significant support from a variety of users for a certain period of time. As far as the scope and time-frame of this study allowed, attempts made to obtain a good level of understanding of the users' experience on the CAM software, covering its multiple aspects. Although the rate of response to the questionnaire was under expectations, it allowed us to go through the details of the responses in order to understand the correlation between responses and the questions and between the responses per se.

Overall, as the first-ever effort in analysing CAM, the joint elicitations of information extracted from interviews and questionnaire provided the CAM support team with a sufficient amount of information to be informed on major comments of a wider community of the CAM users. Perhaps keeping the community of CAM users more engaged in the assessment process might help the possible future investigations of the software. To summarise the observations from the interviews and the questionnaire, the following points can be highlighted for future CAM improvements, as is shown in Table 3.4:

Table 3.4. The summary of the highlights from interviews and survey of CAM

Perspective	Highlights from the interviews	Highlights from the survey
Model building	Tutorial and training material Functionality during the modelling Creating a pallet of OPM	Tutorial and video clips (88%) Generating or modifying new function (41%) Running a simulation (65.4%)
Model simulation and analysis	Interface and tracking mechanism Changing views between toolboxes Partitioning, banding, and clustering	Interface and ease of use (54%) Functionality and performance (34.7%) Data analysis (DSM optimisation) (56.3%)
Model visualization and making report	Compatibility and integration with other software packages Export issues (e.g., from DSM onto PDF) Import issues (e.g., import file from excel)	Compatibility or integration with other software packages (23%) Flexibility in generating output files (37.5%) Organization and documentation of results (19.2%)
General concerns	Installation in different platforms Commercial usage conditions Availability of further plugins	Installation of the software onto the target platform (39%) Configuration of the software following the installation (46%)

Chapter 4

4. A quantitative approach for evaluating CAM

In the previous chapter, an empirical approach presented to investigate the functionality of CAM from the standpoint of its internal and external users. The results however represented that *building a model* and *analysis of output data* have been the most frequent concerns of the users (See Fig. 7.1, in the Appendix). Furthermore, comparing to the other toolboxes, building and simulating a model in the ASM toolbox reported as a major concern, to both internal and external CAM users. It is therefore worth to further investigate the flexibility of the ASM toolbox in supporting multiple project situations.

To deal with these issues, this chapter examines the impact of modelling and simulation functions on design process planning. The specific objectives of this chapter are to (1) understand the basic requirements of a process modelling platform (RQ.4) and to (2) understand the key modelling characteristics of a process model (RQ.5). In fact, the overall goal is to figure out how a single DP can be modelled in a simple variety of ways. This can further help the modeller to understand the minimum levels of information that is required to model a specified project situation, satisfying the targets (Browning, 2010).

In this way, I focus on modelling and simulating different versions of the Signposting system, which is the result of a number of researches that have been undertaken over the years in EDC. The process modelling toolbox (ASM) of the CAM software selected for this purpose. Being applied to a large number of case (mostly in aerospace industry), it has reflected a good capability in building and simulating multiple project types, with different characteristics and at different levels of abstraction.

Therefore, in the following of this chapter, I present an overview of the evolution of Signposting systems over the past 17 years. Amongst multiple versions of Signposting in the history of EDC, I select three of them actually those with the most improvements in their functionality. They are referred to as the *Original* signposting, *Extended* Signposting, and *Applied* Signposting in the rest of this report. After that, I re-build and re-simulate them all in the ASM toolbox of CAM and compare their simulation outputs in terms of the total project duration. The studies in this chapter have been illustrated using the case of a simple Mechanical design process (Clarkson and Hamilton, 2000).

By the end of this chapter, I present further analysis and discussion of the sensitivity of the CAM software in satisfying modelling objectives. The present study, in fact, is the first-ever study of its kind in the history of Signposting systems: a kind of detailed comparison between *information-driven* process models (the Original and Extended Signposting systems) and such *dependency-driven* models (the ASM). The outcome of the study in this chapter will be further discussed in the following chapter to identify directions.

4.1. Evolution of Signposting systems

The literature of Signposting system is tied up with the literature of developing ASM toolbox that has superficially been described in Section 2.2.1. This section presents a more detailed description of the previous research in Signposting over the past 17 years. Figure 4.1 displays an overview of the evolution of Signposting.

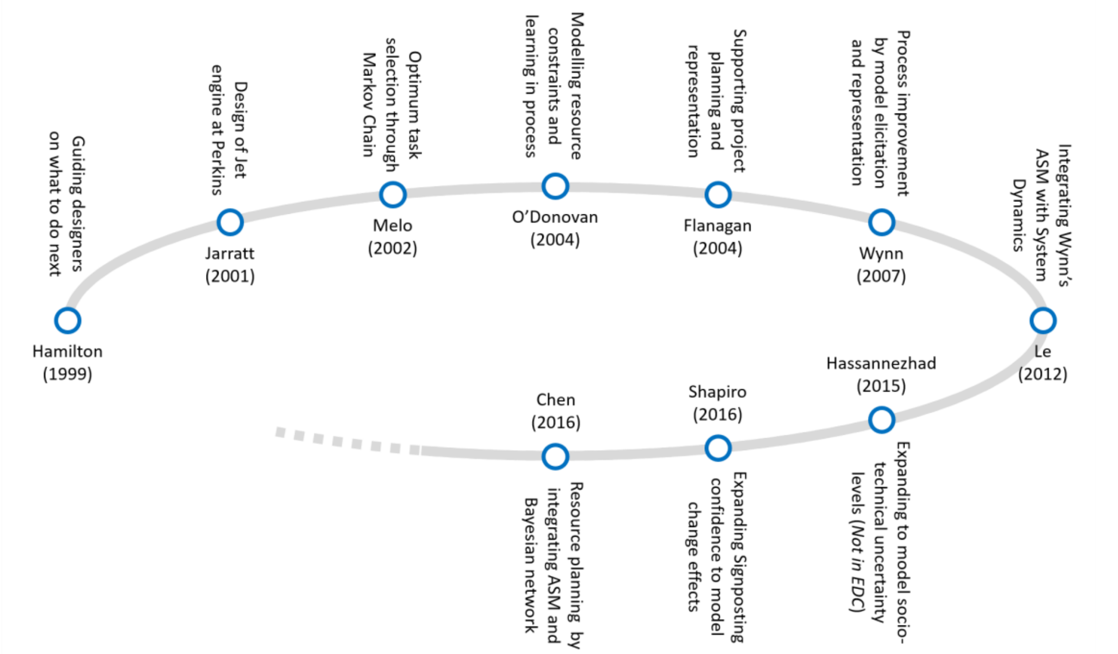


Figure 4.1. The evolution of Signposting systems

Signposting is a dynamic design process modelling approach that can also be used for project planning and simulation. Originally, the signposting approach was developed as a response to the challenge of modelling *helicopter rotor blade* design at Westland to guide designers to the next task, by showing designers those tasks for which they had appropriate input data to advance the state of the design knowledge (Hamilton, 1999). In addition, the technique supports optimum task ordering for the entire project by selecting the most appropriate option from a list of available tasks (Melo, 2002). The technique is versatile and can model project-specific constraints, as well as alternative routes through the DP.

The state of a parameter in Signposting is indicated in terms of the subjective *confidence* that the designer has in its refinement. A task order is implicit in the confidence values and the effect that one task has on another is determined by confidence mapping. Task failure is represented by reduced confidence in the parameter output as opposed to the input and a probability specifying the likelihood that a failure occurs. Partial failure can be modelled by creating a confidence mapping that is lower than required but not below the input level of confidence. However, both kinds of failure drive iteration in the model.

Due to the potential confusion on choosing tasks at any point of the process, given the high risk of rework between possible tasks, an improvement was carried out by Melo (Melo, 2002) by defining an optimum task ordering. This was carried out by selecting the most appropriate option by means of Markov chain analysis, with emphasis on the whole route of the process instead of only the next task. In parallel to the model developed by Melo, Jarrett applied Signposting on the conceptual design of jet engines, by integrating the functionality of Signposting with industrial design tools (Jarrett, 2001).

In order to make Signposting applicable to real cases, an *Extended Signposting* was proposed by O'Donovan (2004). The author attempted to add detailed features to the model, specifically multi-class resource constraints and in-process learning through parameter evolution. As the complement to what Melo was proposed on parameters mapping (Melo and Clarkson 2001), in Extended Signposting, modelling non-Markov processes allowed by dedicating numerical levels to parameters that enables the model to consider all types of real-life parameters.

Moreover, in comparison to previous versions, Extended Signposting could capture multiple possible outputs with different degrees of success, and also estimate the impact of different inputs on possible outputs. Nonetheless, this research was later criticized by

Flanagan (2006) who highlighted two drawbacks of previous work with specific attention on modelling dependencies and parallel tasks. The core of the model was dealing with project planning and representation through investigating the effect of different sources of uncertainties, process properties like scale and connectivity, and the product-process link as interdependencies.

In order to make Signposting applicable to large-size cases and independently from industry type, Wynn (2007) suggested improvements with a focus on elicitation through process modelling and simulation and also model representation by understanding various modes of iteration. He used a sophisticated hierarchical structure of tasks and parameters as a support to model presentation enriched by a user-friendly platform. However, due to dependency-driven nature of model as a task-precedence network, the model would be more suited to categorize as DSM tool rather than Signposting.

After developing the ASM methodology, several studies (at the Cambridge EDC) attempted to sophisticate aspects of model building in ASM in order to enhance its functionality in reflecting real-world process constraints. Examples of studies can be found in Table 2.1. At the same time, outside of EDC, Signposting system has been used as a technique to model aspects of socio-technical uncertainty in engineering design (Hassannezhad, 2015). In his research, the author extended the original Signposting model and combined the activity-based and agent-based concepts of process modelling to additionally consider the allocation of agents to design tasks. Rather than guiding the next task, the objective was to find the right choice of agent for each design task, considering the availability of knowledge and level of expertise (Hassannezhad et al., 2015).

Very recently, and as mentioned before, Chen (2016) presented a method to model different resource types (designers, computational, testing), by combining the ASM model (as a task-based network) with the Bayesian network, and studied the impact of using different options of those resources. Simultaneously, Shapiro (2016) expanded the concept of Signposting confidence mapping in ASM and developed a change management method to enhance the understanding of DP change effects on process performance. The objective was to support process execution through suggesting mitigating reactions to the changes, and support process planning through identifying and prioritising the right changes.

4.2. An overview of the quantitative approach

Supported by Table 4.1, the previous section presented the outline of research in Signposting systems over the past two decades at the Cambridge EDC. In fact, over the time, the research plan has been shifted from identifying and executing the best possible tasks (in order to rationalise the DP) to identifying and executing optimal process planning (in order to prescribe the DP behaviour).

As the result, some versions of Signposting have been mainly concerned with architecting a more detailed process modelling tool, while the focus of the rest was basically on the implementation and applicability of those models with respect to real-world constraints. For the purpose of analysis of functionality of Signposting models in this report, I selected three versions of Signposting with major improvements on modelling characteristics:

1. The original Signposting model (OSM) developed by Hamilton (1999);
2. The extended Signposting model (ESM) developed by O'Donovan (2004), giving major improvements in modelling multiple resource constraints, expanding the concept of confidence, and enabling to model learning during the process;
3. The applied signposting model (ASM) developed by Wynn (2007) with improvements on modelling iterations, information elicitation, and representation.

In the following of this chapter, I first present an overview of the structure of each model by preserving their original building platform. Accordingly, the reader is referred to the original material at each step. After that, I re-simulate these models using the ASM toolbox of the CAM software. Eventually, the analysis of simulation results is presented in that I discuss how different types of DP modelling can be simulated in a single modelling platform and how they behave in terms of more general (i.e., best process plan, robust plan) and more specific (i.e., best task execution policy) process planning issues.

4.3. Building Signposting models

This section presents the mechanism of building Signposting models with respect to their description in the literature. Meanwhile, the architecture of the models is presented with particular attention to the task and parameter mapping.

4.3.1. Original Signposting model

The Original Signposting model – referred to as OSM hereafter – comes out of the realisation that many important design processes have structures that defy conventional linear process descriptions. They involve complex interdependencies between design choices, so that designers have to estimate parameter values, backtrack, and repeat some tasks many times before all the parameters have satisfactory and mutually consistent values, even when what the parameters are is well understood (Stacey et al., 2000).

The model is based upon the assumption that the DP may be thought of as a series of tasks concerned with the identification, estimation, and iterative refinement of key design and performance parameters until a sufficient level of confidence in those parameters is achieved (Clarkson et al., 1999). It starts from the insight that while companies might not understand an overall process, individuals know the tasks that they do and the information that they need, therefore the model of the process is constructed from local understanding. It is based on a task-based representation, meaning that the overall DP is decomposed into simpler and shorter activities (O'Donovan et al., 2004).

Model architecture. A system architecture for the original Signposting version is depicted in Figure 4.2, adapted from Stacey et al. (2000), where build-time functionality is shown on the left, and design-time functionality on the right. According to the figure, key concepts in the signposting approach are a representation of tasks, parameters, confidences, and dependencies. The signposting architecture presented in Fig. 4.2 is summarised in the reference Clarkson and Hamilton (2000) at four levels:

- *Parameter* level, which includes the parameters used to describe the design;
- *Task* level, which includes the tasks available to be used in the design process;
- *Process* level, which organises the tasks;
- *Interface* level, which provides the user with access to the tasks.

As far as related to the model building, the performance of a Signposting system is heavily tied up with an appropriate definition and representation of its tasks and parameters.

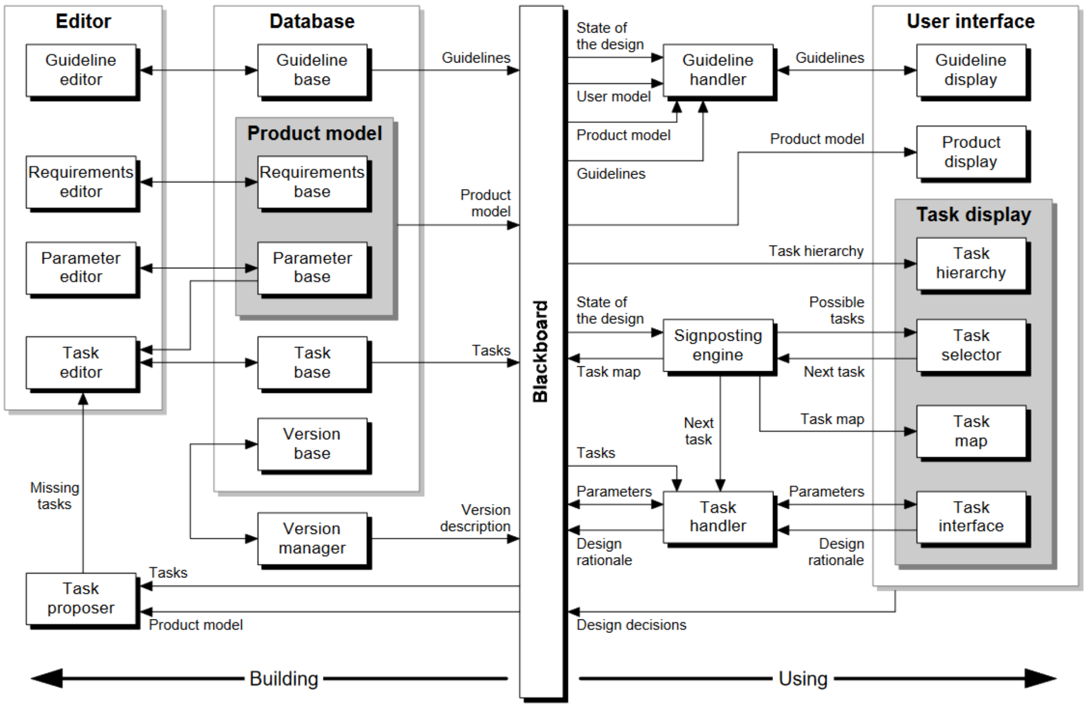


Figure 4.2. The architecture of a Signposting system (adapted from Stacey et al., 2000)

Mapping tasks. A task in Signposting is generally used as the primary building block of the process. It is an activity that takes the values and confidences of certain parameters as inputs and generates or updates the values and confidences of other parameters while the mapping provides a link between the input and output parameter confidences (Stacey et al., 2000). Description of a task in Signposting is presented in Figure 4.3, adapted from Clarkson and Hamilton (2000). It includes confidence matrices (rectangles in the figure), naming their input and output parameters, and describing the confidences required for the input values and expected for the output values.

To perform this role, the generic task representation must satisfy a number of criteria. In particular, it must be applicable at varying degrees of abstraction, appropriate to represent all tasks, linked to the knowledge required to perform the task and represent meta-knowledge specific to the task. The representation must, therefore, couple the knowledge describing the specific method to be used to perform the task with meta-knowledge describing the context in which the task should be performed and the likely consequences of performing the task (see the big rectangle in Fig. 4.3, in the middle).

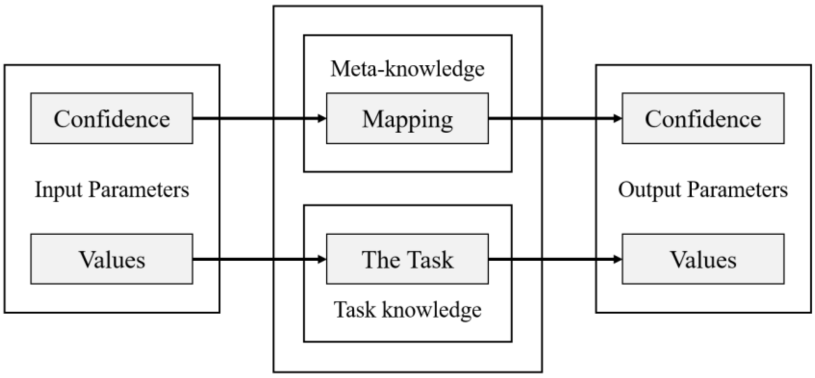


Figure 4.3. Task representation in Signposting (rebuilt from Clarkson and Hamilton, 2000)

Mapping parameters. According to Stacey et al. (2000), a parameter is an aspect of a design that defines the product's physical structure and needs to be determined, and hence embodies a decision about the design. Parameters may have single numerical values, and that their values may be symbolic, or have an internal structure (such as complex shapes) or be clusters of related parameters. In Signposting, the value of a parameter, which is called *Confidence*, is depended on parameters.

Mapping confidence. It encompasses a number of meanings (Clarkson and Hamilton, 2000). To be confident in a parameter means that the parameter is detailed, accurate, robust, well understood, physically realistic and, in the case of a performance parameter, meets pre-defined performance requirements. The confidence in the output parameters is then a function of both the accuracy of the particular task and the confidence in the input parameters (Clarkson et al., 1999). In the original Signposting, confidence is represented using three discrete levels:

- *Low*: assigned to an initial un-proven design or performance estimate;
- *Medium*: assigned to a feasible design or performance estimate;
- *High*: assigned to a feasible design if the resultant product performance satisfies the design requirements.

For more information of the representation of confidences, parameters, and tasks, the reader would refer to the additional material on the subject, such as Hamilton (1999), Clarkson et al. (1999), Clarkson and Hamilton (2000), and Stacey et al. (2000).

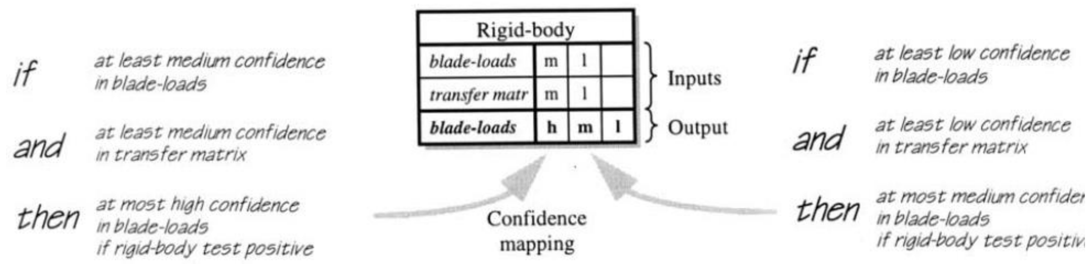


Figure 4.4. Mapping confidence in original Signposting (adapted from Clarkson and Hamilton, 2000)

4.3.2. Extended Signposting model

The Extended Signposting model – referred to ESM hereafter – developed by O'Donovan (2004) improved the original system in several ways, mainly in terms of the expanding the concept of parameter confidences, extending task states, the possibility of modelling multiple-class resource constraints, and in-process learning (basically as the result of parameter evolution). In addition, compared to the previous versions, modelling non-Markov processes was allowed in ESM by dedicating numerical values to the parameters, thus enabling the model to consider all types of real-life parameters. In addition, O'Donovan (2004) used Monte-Carlo simulation to develop a state-contingent plan representation termed the Conditional Precedence (CP) matrix from this model.

Mapping tasks. Tasks in the ESM define transitions from one process state to another through their *input* and *output* states (respectively IN and OUT in Figure 4.5). An *input state* is a process state which describes the minimum levels of parameter qualifier (confidence levels) necessary to execute a task; required information, resources, documentation, for instance. Attached to each input state are values for cost and duration of the task, performed with that level of information and resourcing. An *output state* defines the new process state obtained after execution of a task and representing different degrees of success or different modes of failure (O'Donovan, 2004). Each output state has a probability of occurring, and the sum of all probabilities for the output states attached to each state is one.

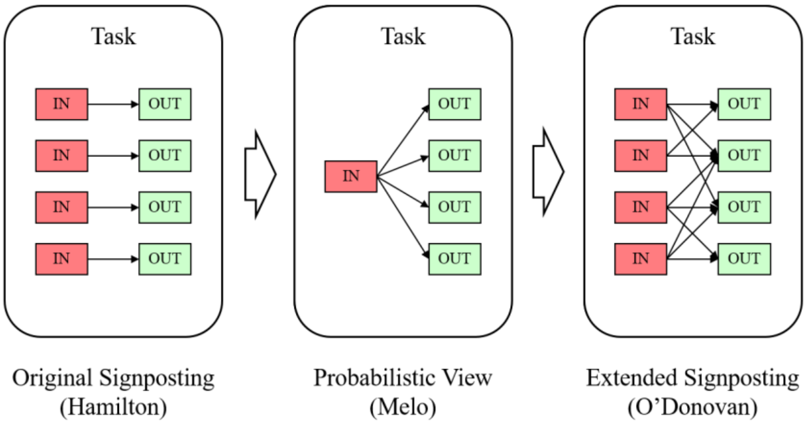


Figure 4.5. Mapping tasks in extended Signposting in comparison with the previous versions (adapted from O'Donovan, 2004)

Regarding Fig. 4.5, in previous versions of signposting, tasks have incorporated multiple input states, each mapping to a corresponding output state (Hamilton, 1999), and single input states mapping to multiple output states (Melo, 2002). The ESM supports both these approaches, with each task having multiple input states mapping to multiple output states. The multiple input states may represent different levels of information input, resource allocation, learning for the task, and even different degrees of success or modes of task failure (O'Donovan, 2004).

Mapping parameters. In the ESM, a parameter is an element of the DP that might be considered as an input to or an output from a task. Comparing to the previous versions, ESM expanded the scope of parameters to include both those conventional elements, such as geometry or electronic component values, and also those less conventional elements, such as manpower or learning during the DP. In fact, this is a significant departure from the earlier versions of signposting, in which a parameter was used primarily in the conventional sense.

It is previously mentioned that Signposting is essentially a parameter-driven approach, in the sense that parameters are the atomic units of the models and tasks, process states, success, or failure of a project, all are described essentially in terms of parameters, or entities composed of parameters. In the OSM, parameters usually referred to elements of the description of the design object. In the ESM, the meaning of a parameter is much more general compared to the previous versions, and it can represent any entity or group of entities in the DP which changes over the course of the design (O'Donovan, 2004).

Mapping confidence. Previous versions of signposting used the term 'confidence', embodying the designer's confidence in a given parameter (Clarkson and Hamilton, 2000). As long as expanding the scope of parameters in the ESM, the value of a parameter has been renamed from *confidence* to the parameter *qualifier*. Some issues relating to the ambiguity and appropriateness of the term confidence has been reported in the literature (O'Donovan, 2004). For example, depending on the type of parameter, confidence can be interpreted in a number of ways.

As the result, in the ESM, each parameter has an *identifier* and a *qualifier*. The identifier is simply the name of the element. The exact meaning of qualifier varies with the type of parameter, but in general, it is an abstract representation of the quality or maturity of the parameter. Except the specialised design tool created by Jarrett (2001), the actual value of a parameter has not been used in Signposting. In ESM, each parameter has at least one qualifier level for each task that creates, modifies, or tests the content of the parameter. The example of qualifier levels for a number of parameters in ESM is presented in Table 4.1.

Table 4.1. Meaning of qualifier in the Extended Signposting (adapted from O'Donovan, 2004)

Qualifier Level	Chassis	Ergonomics (satisfaction)	Ergonomics (accuracy)	User manual	Toaster slot width
0	Undeveloped	No satisfaction of requirement	No knowledge	Not created	Unfixed
1	Concept	Marginal satisfaction	Designer's intuition	Draft	Fixed
2	Detailed design	Complete satisfaction	Computer modelling	Printed	Fixed based on market research
3	Final design	Exceeds requirement	Prototype testing	(not used)	(not used)

In earlier versions of the Signposting system, three levels of qualifier (so as referred to the confidence levels) were used, high, medium, and low. More generally, in ESM, a parameter can have any number of qualifier levels, as many as are needed (see Table 4.1, for different examples). To reflect this, in the current implementations of the model, numerical qualifier values are used, with zero indicating the absence of a parameter, and the positive integers representing progressively higher levels of maturity and quality.

Mapping resource effects. In extended signposting, resources may be treated as another category of the parameter. This has the advantage that no new types of model element are needed since resources can be incorporated simply through the addition of new parameters. The major difference between resource-type parameters and other types of the parameter is seen when considering how resource parameters change during a concurrent process (O'Donovan, 2004). *Resource* here refers to any person or facilities with a finite capacity which participates in the DP (O'Donovan, 2004). This could be for example multiple engineers being assigned to the task, or the distinction between a novice and an experienced engineer (O'Donovan et al., 2003).
The example of modelling resource effects on a Signposting task is displayed in Figure 4.6. Each significant resourcing level can be expressed as an alternative input state for a task, with differences in time, cost, the probability of success and even the types of outcomes possible (Fig. 4.6, top-right). The parameter qualifier for a resource-type parameter can be interpreted simply as the number of units of that resource, whether that parameter refers to the teams or people.

Mapping learning effects. This has been defined in O'Donovan (2004) as follow: "Knowledge or understanding relating to a task or tasks generated within the timescale of the process, which has a significant impact on the speed, cost or success probability of a task or tasks which are repeated within the process, but which is not essential for the execution of the task." This was not a good match for the observed process in the industry, where design iterations are normally used to learn about possible solutions and to converge towards a final design.
The previous versions of Signposting did not model learning during the process. This meant that for any task in the DP, the likelihood of success for the first time that the task is

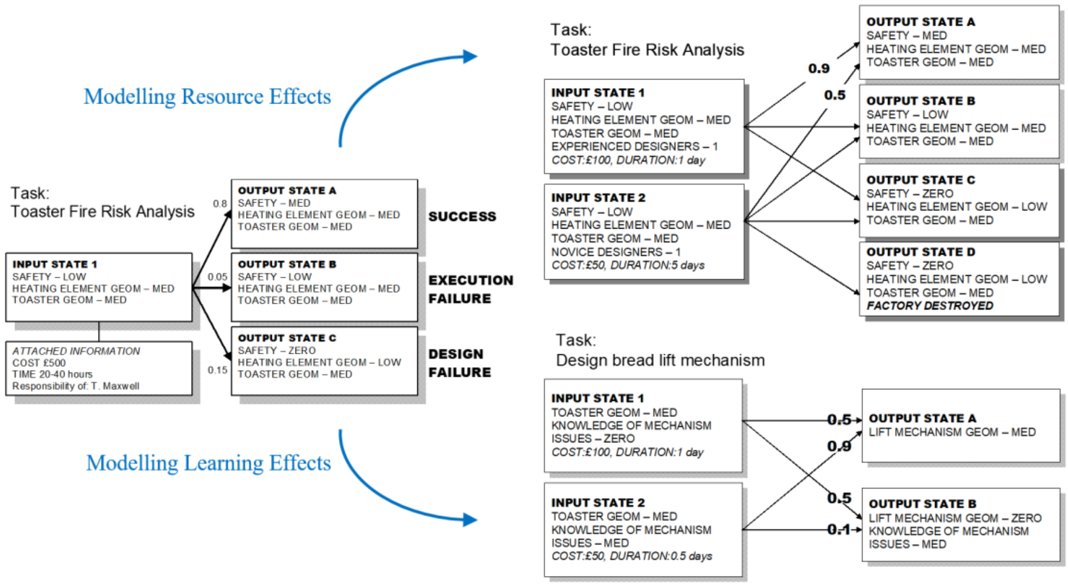


Figure 4.6. Modelling resource and learning effects in Extended Signposting (individual images are adapted from O'Donovan et al., 2003)

executed was the same as the chance of success on each subsequent execution. In the case of a single task which iterates locally until completed, or an iterative loop of tasks, this leads to a characteristic exponentially decaying probability distribution for the number of task or loop executions (Melo, 2002).

Task experience effects may be incorporated into the signposting model through the addition of a new class of parameter, the *task learning* parameter, and the addition of new mappings to tasks to incorporate the learning parameters (Fig. 4.6, bottom-right). *Task learning* parameters can be treated the same way as the other parameters, with the parameter qualifier interpreted as an aggregation of the quality and quantity of learning about a specific design activity or aspect of the product.

For a more detailed explanation of multiple aspects of modelling in the ESM, the reader would refer to the references O'Donovan et al. (2003), O'Donovan (2004), and O'Donovan et al. (2004).

4.3.3. Applied Signposting model

The ASM is a model-based framework to support planning practice in aerospace design. It is a precedence-based modelling framework aiming to support capture of DPs in terms of tasks and their dependencies (interactions). The idea was that a design task might involve consideration of a great deal of information, while only a small number of parameters are usually considered to drive the ordering and selection of tasks (Wynn, 2007).

The earlier versions of the model considered activities and iteration as the main components of project plans (Wynn et al., 2005). In particular, application of the approach to model processes at a more detailed level has necessitated a more sophisticated representation of hierarchies and of the role of information in driving process behaviour.

Similar to the previous versions of Signposting, the approach characterizes designing as the estimation and refinement of parameters, where the term parameter can refer to any aspect of the product or process which may change during design (Wynn et al., 2006). The approach considers that DP is driven in part by changes in the availability and state of design parameters. An overview of the ASM modelling framework is displayed in Figure 4.7, adapted from Wynn et al. (2006).

Mapping tasks. Similar to the ESM, tasks in ASM are associated with states, in the sense that state refers to an abstract description of the current value of a parameter. It means that a task cannot be undertaken unless the required parameters are available, perhaps in a specified state (Wynn et al., 2006). In fact, tasks in ASM cause parameters to become available, and/or the state of parameters to change.

The ASM provides three classes of task (see Figure 4.8 for example of each):

- *Simple* tasks, which have one input and one output scenario;
- *Iteration* construct, which has one input and two output scenarios;
- *Compound* tasks, which have one input and any number of output scenarios.

The simple task is used to model tasks whose execution is not considered to immediately affect process routes, such as data file conversion tasks. The compound task is a general-purpose element used to represent any activity which may include a decision or outcome affecting the choice of next task (Wynn, 2007).

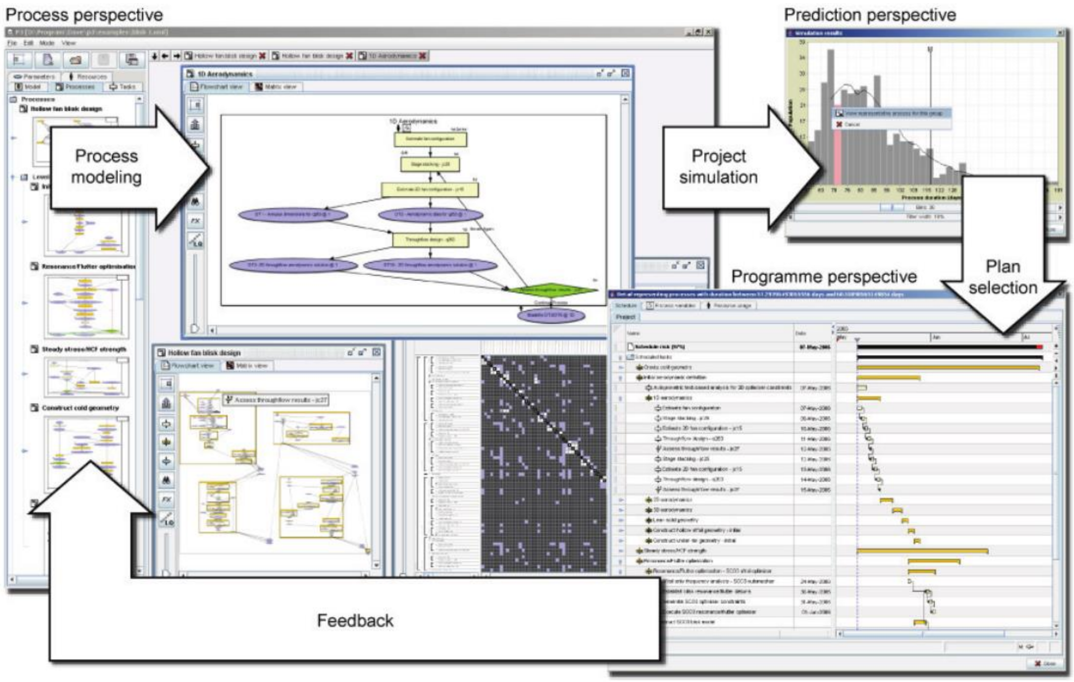


Figure 4.7. An overview of modelling and simulation in ASM using the original P3 Signposting software (adapted from Wynn et al., 2006)

The state of each task is determined by examining the information state of the model and the required input information for that task. Due to the hierarchical definition (unlike the previous versions of Signposting), only the state of information in the task's parent process can affect its state. Processes are therefore described in terms of knowledge about individual tasks and their input/output characteristics, and assumptions regarding the limited scope of a task's effect upon other tasks in the model (Wynn, 2007).

Mapping interactions. The ASM framework captures dependencies and precedence between tasks in terms of their interactions with parameters. Two classes of interaction are introduced in the model and are used to distinguish the role which the interaction is considered to play in driving process behaviour (see Figure 4.9 for examples of interactions in ASM):

- *Data interactions* used to represent the requirement or production of information by a design task in the weaker dependency form which does not directly drive task selection.

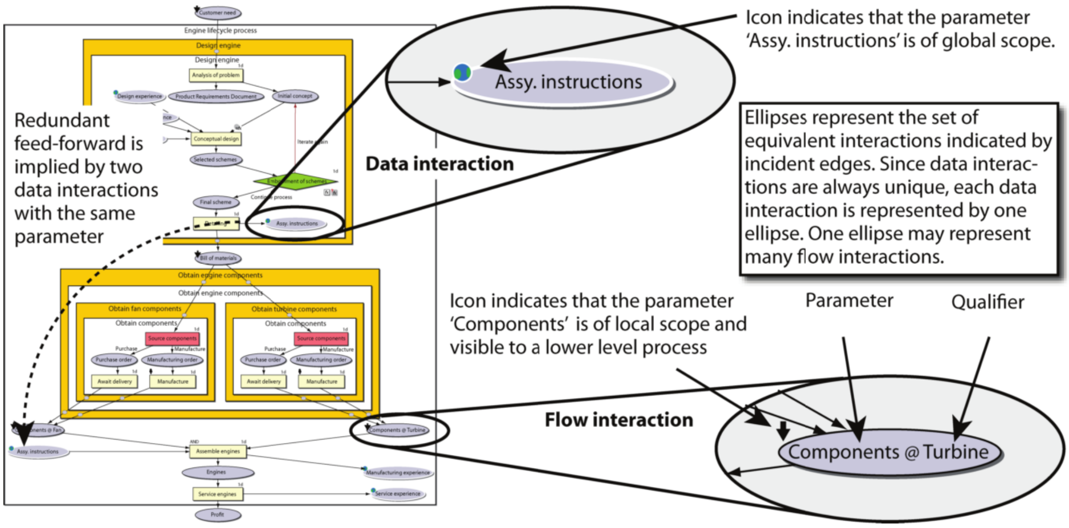


Figure 4.9. Two classes of interactions in ASM (adapted from Wynn, 2007)

- *Flow interactions* used to represent the stronger precedence requirement for a parameter to be updated prior to attempting a task.

Mapping parameters. Parameters in ASM might be referred to the direct representation of a design object (e.g., the diameter of a turbine blade cooling passage), or to a data file which produces a report to satisfy a design review or a course of action, the design team intends to take. In addition, interaction qualifiers are used to indicate the state of a parameter (Wynn et al., 2006).

In the ASM, state refers to an abstract description of the current value of a parameter. Changes in the state during a process indicate that a parameter may play a different role in task selection and execution as the design progresses. The precise definition of a state is *context-dependent*. It might be used to represent a set of indeterminate components; the qualifier indicates precisely which parts are represented in the context of a particular set of interactions, or might be used to represent specific design information, for instance, indicating the three levels of confidence or maturity of the design scheme (Wynn, 2007).

Mapping confidence. O'Donovan (2004) used the term 'qualifier' due to the potential confusions that the term confidence might have been created. In the ASM, the framework again incorporated the concept of confidence to reflect the degree of maturity of information in the modelling. According to Wynn et al. (2006), it is introduced as an abstract quality which may take a number of meanings, typically referring to the designers' belief in their solution or some other aspect of design maturity. It may be described qualitatively using interaction qualifiers or quantitatively using process variables.
As the result, as the DP progresses, levels of confidence are considered to increase non-monotonically. However, it often occurs that an evaluation activity reveals the previous shortcomings in the design. Such activities can be modelled as tasks which may reduce levels of confidence. This reduction may, in turn, necessitate the rework of tasks which have already been completed (Wynn, 2007).

Mapping resources. The ASM framework allows specification of resource requirements for each task. Each resource requirement indicates a number of units of a resource which the task requires to execute. Resource elements consist of an availability profile, describing how many units are available between given dates, and a calendar, indicating the hours and days for which the resource is available (Wynn et al., 2006). The simulation algorithm assumes that a task cannot begin until the specified units of all required resources are available in the pool. These resources are removed from the pool during execution of the task. Subject to other aspects of model configuration, resource constraints may thus affect simulation behaviour.
Research in ASM has been much more extensive than the other versions of Signposting. So the reader can find more information and insights on development and applications of the ASM framework in multiple contexts in the references such as Wynn et al. (2005), Wynn et al. (2006), Wynn (2007), Chalupnik et al. (2007), Kerley et al. (2011), Le et al. (2012), Shapiro (2016), among many others.

4.3.4. Summary of Signposting models

This section presented an overview the three versions of Signposting model, relying on the core literature, with specific attention on the way that their core elements (tasks, parameters, and confidence) have been modelled in the original frameworks. In fact, it was a more detailed review of the related literature so far, in order to understand how Signposting has been improved over time. The following Table 4.2 summarises the discussion by outlining the main characteristics of the three Signposting versions.

Table 4.2. Comparing the main characteristics of Signposting models

	Original Signposting	Extended Signposting	Applied Signposting
Process mapping	Not allowed	Allowed	Allowed
Task mapping	Multiple inputs Single output	Multiple inputs Multiple outputs	Multiple inputs Multiple outputs
Parameter mapping	Qualitative parameters	Qualitative parameters Quantitative parameters	Qualitative parameters Quantitative parameters
Confidence mapping	Three qualifier levels	Numerical qualifier levels	Numerical qualifier levels
Resource mapping	Designer's confidence	Resource-type parameters	Resource-type parameters
Learning mapping	Not allowed	Allowed	Allowed
Dependency mapping	Task-parameter dependency within a task	Information dependency between tasks	Precedence and dependency in terms of interactions between tasks and parameters

4.4. Simulating Signposting models

Established on the information outlined in the previous section, this section re-simulates the three Signposting versions by using the ASM toolbox of CAM. It was a very critical task, in fact. A significant amount of effort was required by the author of this research to carefully study the relevant literature and understand the models in a way that have been originally developed before. An additional challenge was pertaining to the fact that two of these models, the OSM and the ESM, have not been simulated in the CAM software platform before. However, this was one of the purposes of this study to investigate the functionality of CAM in modelling approaches that CAM has not basically developed for that purpose.

To re-simulate, the models, a simple and generic example of the mechanical component design process (Clarkson et al., 2000) used to examine the functionality of CAM software to support process planning. The model has previously been used in Cambridge EDC in a number of studies as a laboratory experiment. For example, Chalupnik et al. (2007) used mechanical component design to examine the usefulness of an (as is so-called) *one-factor-at-a-time* analysis of process duration. The objective was to study how the ASM modelling framework can be used to evaluate process robustness.

The rest of this section represents a brief description of the use-case, followed by simulation and visualisation of Signposting models.

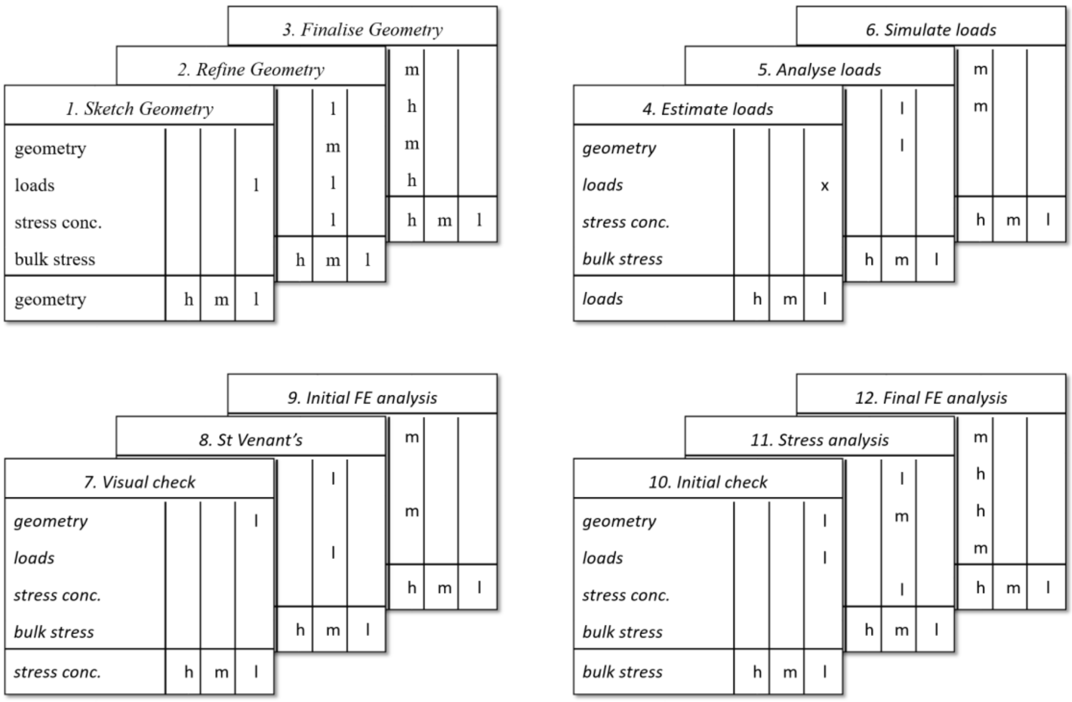


Figure 4.10. The mechanical design tasks and their associated parameters (adapted from Clarkson et al., 2000)

4.4.1. Example case: Signposting mechanical component design

In general, many mechanical component DP can be described (in simplistic terms) as "the definition of a geometry to carry a given set of loads subject to constraints on the allowable bulk stress within the component and on local stress concentrations (Clarkson et al., 2000). In spite of its simplicity, this definition can show the interdependencies between design parameters, which can be the case of so many real design problems.

The example used for the simulation analysis in this research is composed of 12 tasks. They are represented in Figure 4.10. The figure also shows the parameters associated with each task including the minimum input and maximum confidence levels. Accordingly, the task network representation of the example is demonstrated in Figure 4.11. The ticks below the diagonal represent the forward flows and those ticks above the diagonal shows the feedback flows. Clarkson et al. (2000) have noted further process characteristics of this case.

	1	2	3	4	5	6	7	8	9	10	11	12
1				✓								
2	✓				✓		✓			✓		
3		✓				✓		✓				✓
4												
5	✓			✓								
6		✓			✓							
7	✓											
8	✓						✓					
9		✓						✓				
10	✓			✓								
11	✓				✓					✓		
12		✓				✓			✓		✓	

Figure 4.11. The information dependency between the mechanical design tasks

4.4.2. Setting up the simulation

The tasks in the simulation models are characterised in their simplest form so that can enable a better comparison between Signposting models. The duration of each task is characterised using a *triangular probability density function* (TPDF). A single rework cycle is included in the model.

However, the number of *iterations* vary according to the different versions of Signposting models, in the sense that iteration in the OSM and ASM is determined by checking the output confidence levels against the minimum required levels, while in the ESM, it is based on the *probability* of rework on each attempt (similar strategy to the O'Donovan's model). In the model configuration, there are no resource constraints that might limit concurrency, but two types of *novice* and *expert* resource with different knowledge and expertise on the process are considered in the simulation model. Generally, it is assumed that the probability of rework, amount of cost per each task, and the task duration probability density functions remain constant, irrespective of the number of iterations which are attempted (i.e., no learning effects).

However, depending on the type of resource, each task in the models is characterised by different values of cost, duration, and probability of success in the outputs. The summary of simulation characteristics for different versions of Signposting is presented in the following Table 4.3. All the simulations have been accomplished in the ASM toolbox of the CAM software. As mentioned before, the author used the Novice resource-type as the default scenario for all three versions of Signposting. The following section presents the simulation models.

Table 4.3. Setting up the simulation of Signposting models

Modelling elements	Original Signposting	Extended Signposting	Applied Signposting
Original task duration	Fixed TPDF, different for expert and novice	Fixed TPDF, different for expert and novice	Fixed TPDF, different for expert and novice
Rework task duration	Fixed TPDF, different for expert and novice	Fixed TPDF, different for expert and novice	Fixed TPDF, different for expert and novice
Task cost	Fixed rate, but different for each task and for each resource type	Fixed rate, but different for each task and for each resource type	Fixed rate, but different for each task and for each resource type
Dependency between tasks	Based on paramter confidence levels	Precedence network of tasks	Precedence network of tasks
Resource type	Novice / Expert Designer	Novice / Expert Designer	Novice / Expert Designer
Task execution policy	Checking input confidence level	Checking input confidence level	Checking input confidence level
Iteration policy	Checking output confidence level	Fixed probability of rework for each task	checking output confidence level
Learning in process	Fixed duration rate	Fixed duration rate	Fixed duration rate

4.4.3. The Signposting simulation models

The Figure 4.12 indicates the different levels of simulating the OSM, as an example, in the ASM toolbox. See the Section 2.2 and Table 2.2, for more information on modelling in ASM toolbox. The task in the Fig. 4.12 are modelled exactly based on the raw data from Fig. 4.10 (compare the inputs and outputs of each task in Fig. 4.12(b) with the associated task box in Fig. 4.10).

As is shown in the figure, the original version of signposting system considers the construction of a DP from knowledge of *individual design tasks*, i.e., each individual task in Fig. 4.12(b) seems to be a unique meta-model without a direct dependency on other tasks. In fact, modelling tasks in the OSM is associated with eliciting the required knowledge (e.g., in terms of parameter requirements) from the experts. As the result, the model does not capture any explicit knowledge for defining the DP (Clarkson and Hamilton, 2000).

Characterising each in the model is accomplished by double-clicking the task rectangle (for example, task 1 in Fig. 4.12(b)). Doing so, a new task information page will be appeared like what in Fig. 4.12(c) that includes two main panels: description and behaviour. The former contains the basic information on labelling the task, and the latter behaviour (as is shown in the figure) is composed of several sub-panels, such as pre-condition, resources, pre-process, duration, outcome, post-process, variables, and actions, each of which is responsible for specific aspects of the task. The functionality of any of these sub-panels is presented in Table 2.2 when presenting functionality of the ASM toolbox.

Considering the previous discussion in Section 4.3 on building the Signposting models and the simulation process explained before in this section, the author simulated the two other Signposting versions (the ESM and the ASM frameworks) in the ASM toolbox. The screenshots of the simulation models are respectively displayed in Figure 4.13 and 4.14.

Finalising these simulation models and their verification took almost six months of the author with many iterations. During this time, the author conducted many individual discussions with the group members in Cambridge EDC as well as revisiting the original models. An additional challenge was related to the fact that except the ASM framework (which was developed in the ASM toolbox), the two other models have been simulated in other platforms with different settings.

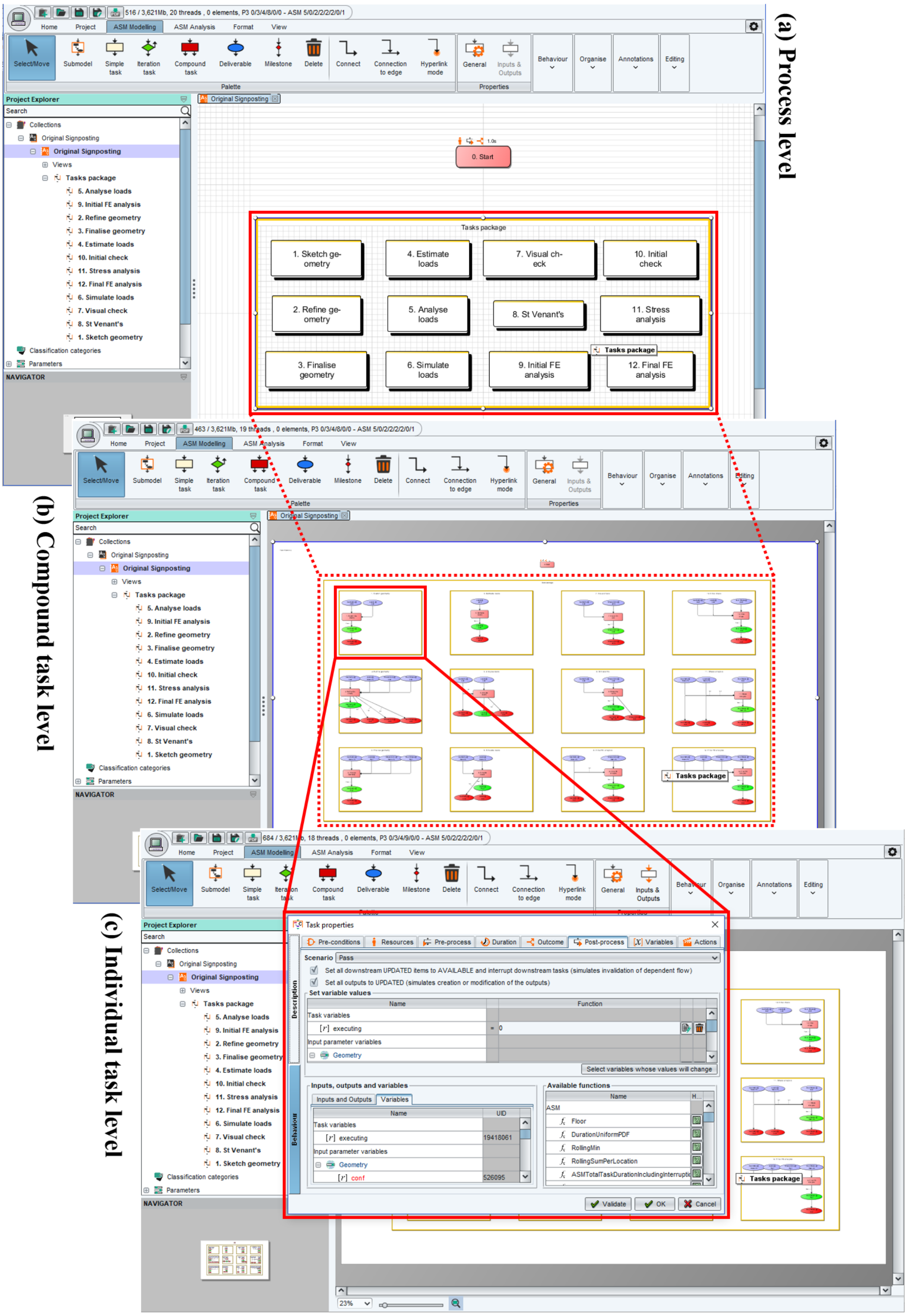


Figure 4.12. The Original Signposting Model reconstructed from Clarkson and Hamilton (2000)

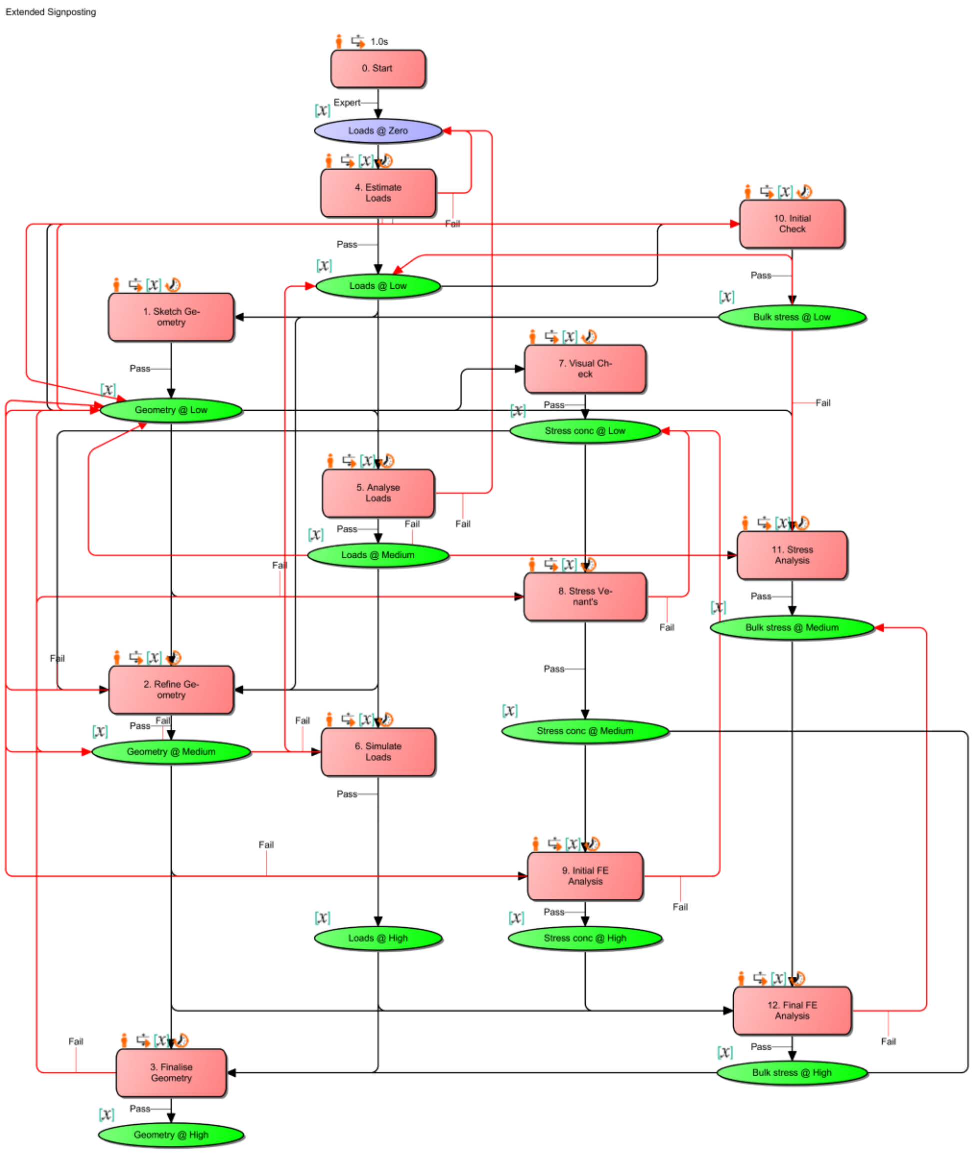


Figure 4.13. The Extended Signposting Model re-constructed from O'Donovan (2004)

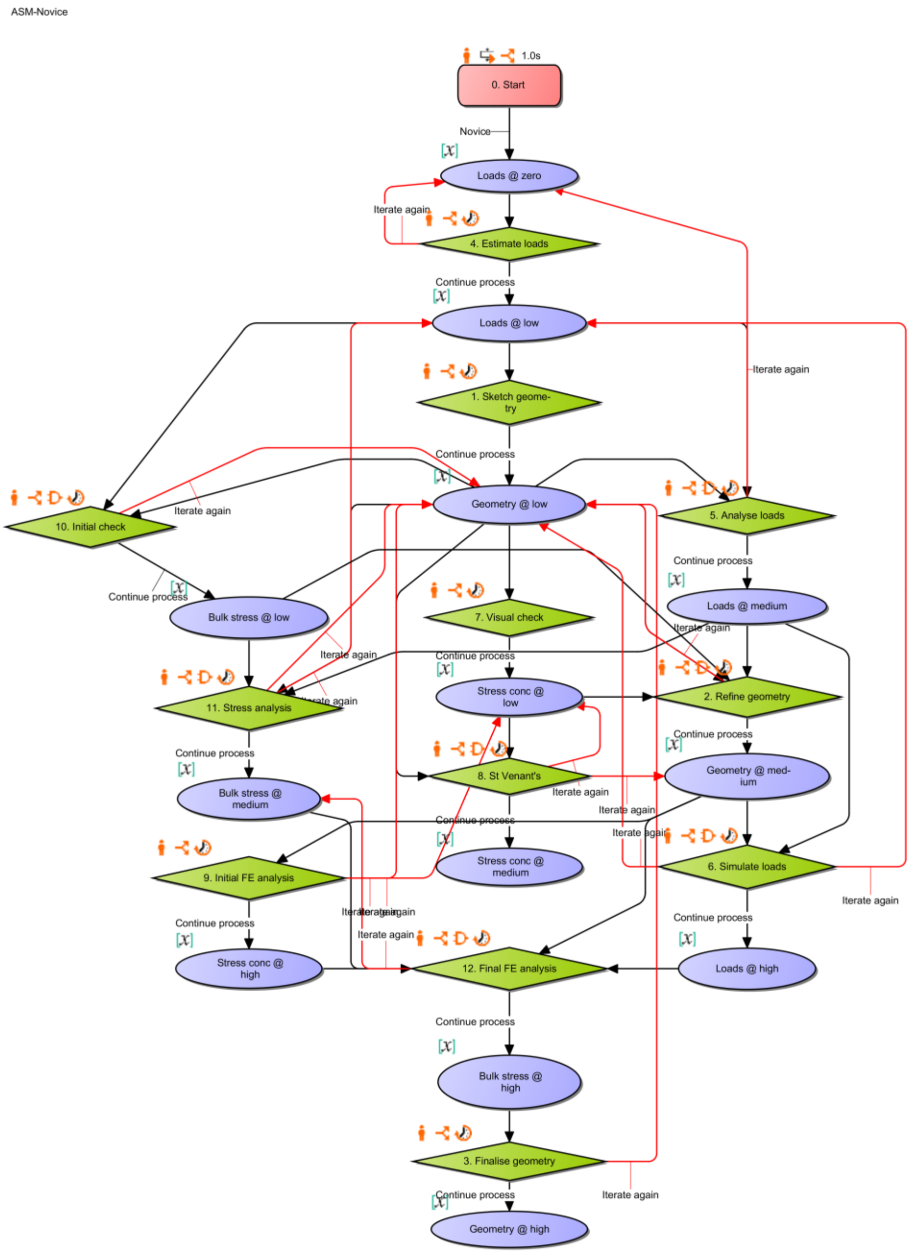


Figure 4.14. The Applied Signposting Model - ASM, re-constructed from Wynn (2007)

The graphical interface of the original Signposting was developed using the *MetaCard development suite* (Clarkson and Hamilton, 2000). O'Donovan (2004) used C++ programming language to develop his Monte-Carlo simulation model of extended Signposting. The black lines in Figures 4.12 and 4.13 represents the forward information flow and the red lines have been used to display feedback (iteration) flows. The result of simulating Signposting models shown in the section will be visualised in the next section and followed by discussion and further analysis of sensitivity.

4.5. Visualising simulation results

After finalising the simulation configurations, the author ran the Monte-Carlo simulation with 1000 replications. The primary results of experiments are presented in Figure 4.15. There are a number of ways to visualise and interpret the simulation results in CAM. The Fig. 4.15 shows the frequency of total DP duration for three Signposting versions. The objective is to compare the functionality of different process modelling framework in addressing the same problem (here the mechanical design case). Another objective is to investigate the functionality of CAM toolboxes in addressing different process modelling characteristics (such as mapping task, parameter, dependency, interaction, resource, etc.).

4.5.1. Identifying best process plan

One of the major attributes of DP plans is the total duration. According to the duration histograms in Fig. 4.15, the ASM represented the total duration within the range of (0, 4) days. This range for the OSM and the ESM was similarly between the range of (0, 2.5) days. In terms of the accuracy of results, it can be implied that the original and Extended versions have reflected a better performance. However, looking into the data in CAM shows that, in spite of getting less variance, the total duration in the original versions are placed at a higher level, when comparing to the ESM and ASM.

The histograms displayed in Fig. 4.15 nevertheless represents the normality of the simulation results. Hence, to get a more detailed information, the simulation results in CAM were exported as CSV files and then, the *Gaussian Function* was computed for each modelling version – it has widely been used in statistics to describe the normal distributions.

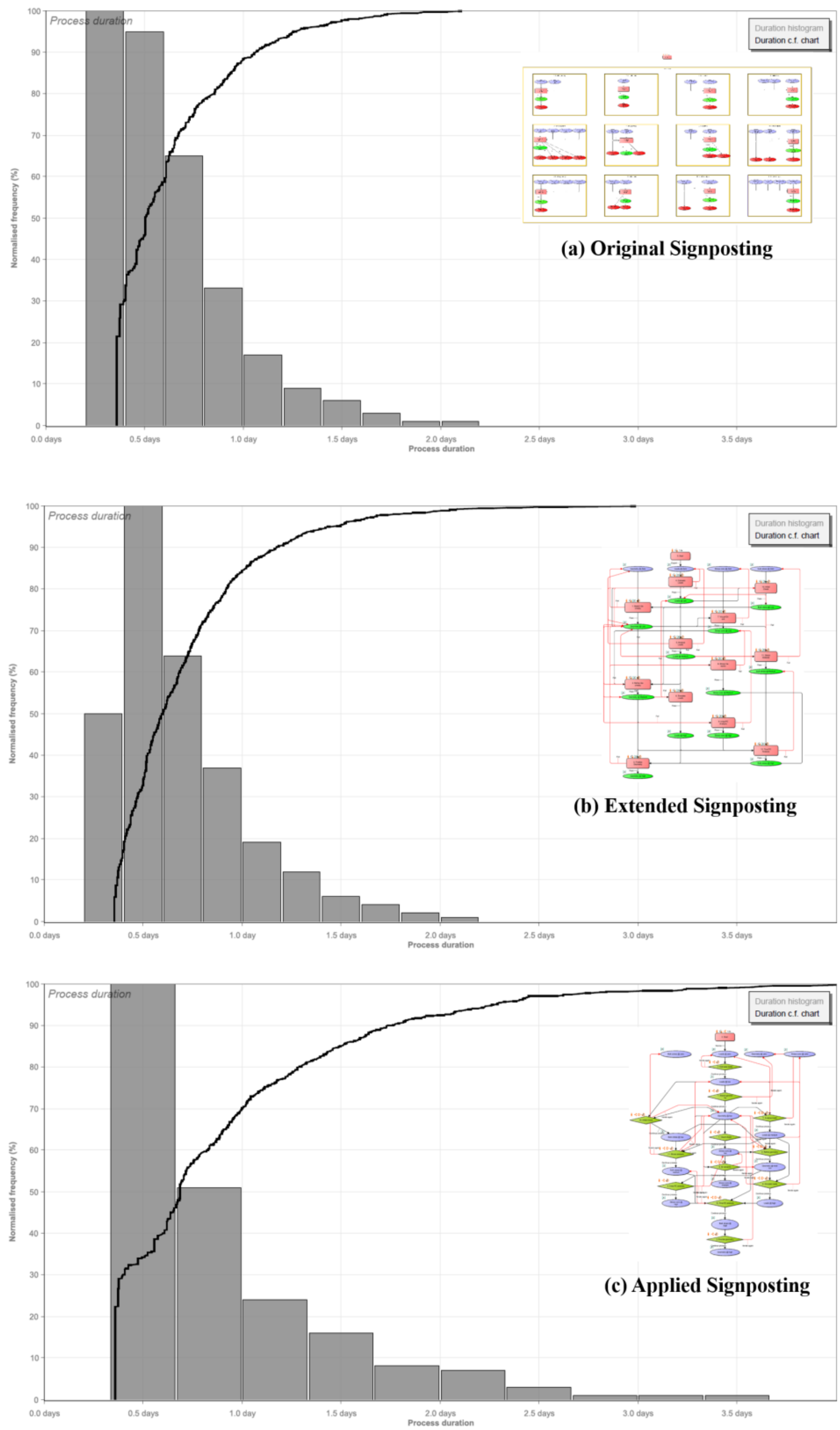


Figure 4.15. Histogram of process duration for different Signposting models

This process is illustrated in Figure 4.16, along with the distribution of outcomes based on different process models. The functions have been computed by the following formula:

$$f(x) = ae^{-(x-b)^2/2c^2}, \quad \forall\, a = {}^{1}/_{\sigma\sqrt{2\pi}}\,, b = \mu\,, c = \sigma \qquad \text{Equation (4.1)}$$

The comparative Gaussian chart in Fig. 4.16 can confirm the primary observations; the OSM has given a result with less variance but with a higher mean, and the ESM and the ASM have respectively given less duration mean but with a higher variance. The obvious point nevertheless is that the results obtained from the OSM have some distance with those

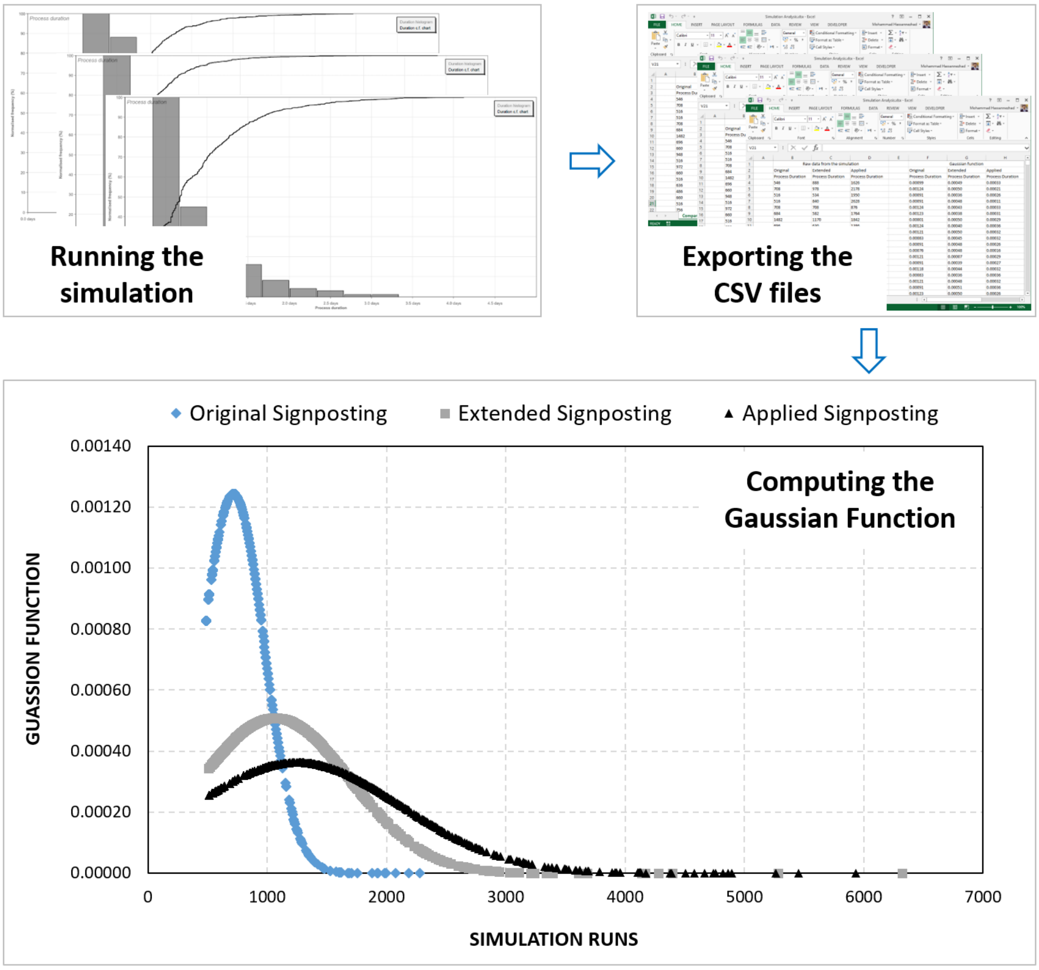


Figure 4.16. Obtaining the Gaussian Function of process duration for Signposting **models**

from its extensions. It might be due to the fact that tasks in the original version are related to each other through their associated parameters, or perhaps because of the higher number of iterations in the original version. These issues are addressed in the following by looking at the best process routes.

Nevertheless, identifying the best process plan depends on the process objectives. If the goal is to minimise the variance, then the OSM might be the best scenario. But if the goal is to minimise the process duration, then either of the ESM and the ASM can be considered. A number of similar studies though in the past attempted to understand the factors influencing best process plan in Signposting, such as O'Donovan et al. (2004) and Chalupnik et al. (2008).

4.5.2. Identifying best process route

Identifying the best process route is a difficult task and several studies in the literature of Signposting can be found relevant, such as Clarkson et al. (2000), Clarkson, Melo, et al. (2001), Melo (2002), and Keller et al. (2006). In fact, it depends on the perspective: to be seen from the perspectives of *critical* tasks or *non-critical* tasks in the process.

Another visualisation aspect of CAM is its single and *probabilistic Gantt charts*. They are accessible by right-clicking on the duration histogram. The number of process routes is, in fact, equal to the number of simulations runs; in our experiment, there are 1000 different routes. In this situation, the probabilistic Gantt chart can help understand the overall shape of optimal process route. The probabilistic Gantt chart associated with each of the Signposting models is presented in Figure 4.17.

The green bars show the first attempt of each task. The red bars show where a task has been revisited. The *density* of colour shows the probability that a task will be in execution at a given time. Further information on interpretation of Gantt charts in CAM can be found in the main software website, within the process modelling sub-heading.

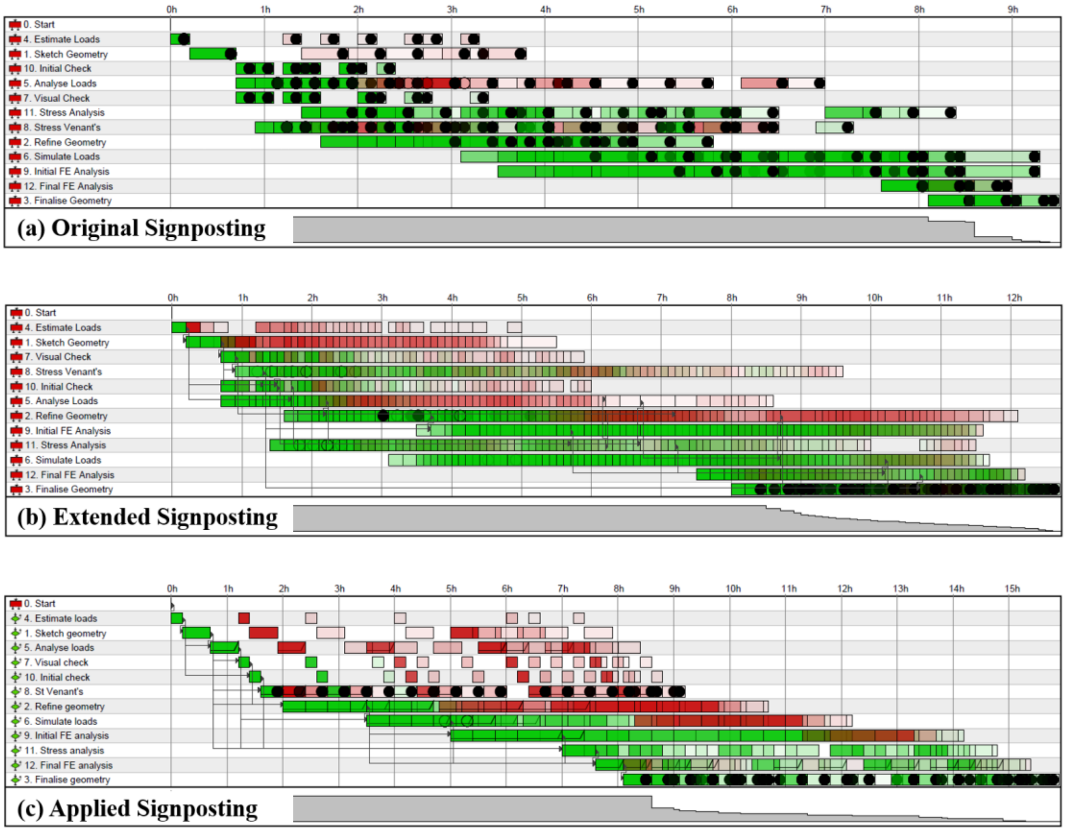


Figure 4.17. Process Gantt charts associated with different Signposting models

These charts are achieved from 386 process routes (out of 1000 runs) for OSM, 381 out of 1000 for ESM, and 430 out of 1000 for ASM, respectively. However, these numbers are based on a one-time 1000-run simulation and therefore, due to the nature of Monte-Carlo simulations, the exact distribution of process routes might be different at each simulation experiment.

By looking at the Gantt charts individually, it is though difficult to realise the most feasible (near optimal) process route. To make a better comparison, the partitioned DSM of the example case is provided in Figure 4.18, right-hand side. If I consider the partitioned DSM as the evaluation criteria for comparing Signposting models, it can result that the ASM framework has reflected a better sequencing of design tasks. After that, the original and the extended versions can be placed at the lower levels.

	1	2	3	4	5	6	7	8	9	10	11	12
1				✓								
2	✓				✓		✓			✓		
3		✓				✓		✓				✓
4												
5	✓			✓								
6		✓			✓							
7	✓											
8	✓						✓					
9		✓						✓				
10	✓			✓								
11	✓				✓					✓		
12		✓				✓			✓		✓	

	4	1	5	7	10	8	2	6	9	11	12	3
4												
1	✓											
5	✓	✓										
7		✓										
10	✓	✓										
8		✓		✓								
2		✓	✓	✓	✓							
6			✓				✓					
9						✓	✓					
11		✓	✓		✓							
12							✓	✓	✓	✓		
3						✓	✓	✓			✓	

Figure 4.18. The original (left) versus partitioned (right) DSM of the mechanical design case

Moreover, obtaining the (near) optimal process route should be considered together with other modelling characteristics and task execution policies, such as the distribution of duration for each task, the flow of dependencies and iterations, level of knowledge (expertise) on each task, and the pass/fail conditions of each task. This sort of experiments requires much further scenario analysis on the simulations, in order to understand the reliability and robustness of process modelling tools, which is out of the scope of this research.

In the following of this section, the author examines the sensitivity of simulation results with respect to the level of expertise in order to provide a baseline for identifying best task execution policy. The rest of above-mentioned experiments can be suggested for future research.

4.5.3. Identifying best execution policy

As discussed before, execution of a design task depends on a number of elements. One of them is relating to the *understanding of the most appropriate level of expertise for each task*. In the ASM toolbox, this sort of analysis requires re-running the model using different resource types (an indicator of levels of expertise).

The only resource to execute tasks in the mechanical component design is the designer, and in terms of the level of knowledge (expertise) on the task, the designer can be *Novice* or *Expert*. These levels of expertise accordingly represent different characteristics during a task execution, in terms of the duration of the task, cost of performing a task, and degree of success, which can be determined by the probability of iteration or output confidence of a task.

The default case of expertise level until now was doing tasks using a novice designer. This cannot be always the case. Therefore, in order to understand the impact of knowledge level (expertise) on task execution, I designed a set of experiments to perform each of the 12 tasks using a novice or expert designer. For this purpose, the ASM framework has been selected as the simulation platform to apply the scenario analysis.

Table 4.4 presents the set of 26 experiments, amongst so many possible scenarios. The first scenario in the table (doing all the tasks using a novice designer) has previously been considered in this section when comparing different Signposting models. I keep that scenario at this step for the purpose of comparison. For the other scenarios, the task information has been modified accordingly and I ran the ASM simulation model with 1000 replications. The presumption was that some tasks might reflect a more sensitivity with respect to the expertise level than the other tasks. This sort of sensitivity analysis helps to understand the most feasible task execution policy.

For visualising the result of scenario analysis, I used the same approach as I previously used in the sub-section 4.5.1 for investigating the best process plan. Doing so, I ran the simulation and exported the results into a CSV file. Then, I computed the *Gaussian Function* for each scenario (to normalise the data and make them comparable) and visualise them comparatively in the chart. The result is shown in the following Figure 4.19.

Table 4.4. Designing a range of simulation experiments to examine the impact of designer's expertise on the task execution

Scenario	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10	Task 11	Task 12
1	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.
2	Exp.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.
3	Nov.	Exp.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.
4	Nov.	Nov.	Exp.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.
5	Nov.	Nov.	Nov.	Exp.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.
6	Nov.	Nov.	Nov.	Nov.	Exp.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.
7	Nov.	Nov.	Nov.	Nov.	Nov.	Exp.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.
8	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Exp.	Nov.	Nov.	Nov.	Nov.	Nov.
9	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Exp.	Nov.	Nov.	Nov.	Nov.
10	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Exp.	Nov.	Nov.	Nov.
11	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Exp.	Nov.	Nov.
12	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Exp.	Nov.
13	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Nov.	Exp.
14	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.
15	Nov.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.
16	Exp.	Nov.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.
17	Exp.	Exp.	Nov.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.
18	Exp.	Exp.	Exp.	Nov.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.
19	Exp.	Exp.	Exp.	Exp.	Nov.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.
20	Exp.	Exp.	Exp.	Exp.	Exp.	Nov.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.
21	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Nov.	Exp.	Exp.	Exp.	Exp.	Exp.
22	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Nov.	Exp.	Exp.	Exp.	Exp.
23	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Nov.	Exp.	Exp.	Exp.
24	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Nov.	Exp.	Exp.
25	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Nov.	Exp.
26	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Nov.

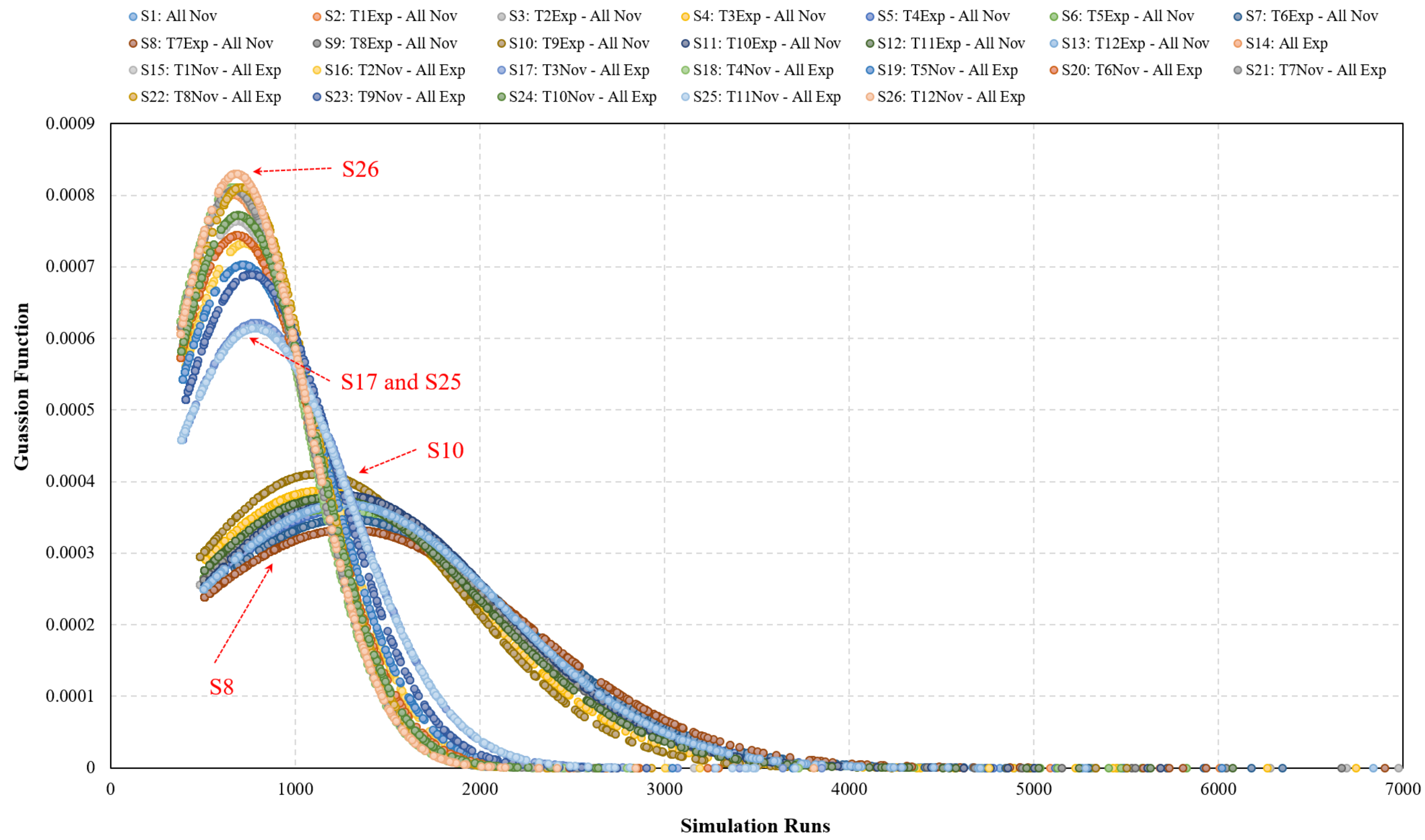


Figure 4.19. Identifying the best execution policy: Impact of expertise level on process execution

Figure 4.19 represents the Gaussian duration distribution for a range of 26 experiments. Two of the experiments (scenarios) are related to executing all tasks using a novice (Scenario 1) or an expert (Scenario 14). The other scenarios can be placed somewhere in the middle in the sense that the expertise level of only one task in each scenario has been changed. The objective was to understand how changing the expertise level in a task can affect process duration and as the result. As the result, his can help understand the criticality of each task in some way related to the expertise level and eventually, help identify best task execution policy (for each task).

Considering Fig. 4.19, at the first look, two general patterns can be determined: the one with higher process duration mean and less variance, and the one with less process duration mean yet with a higher variance in the results. In fact, the first patterns belong to the second half of scenarios where the entire task are executed using an expert designer. On the contrary, using novice designer has shown a shorter but a wider shape of process duration, which means less mean and more variance.

Nevertheless, a few points are highlighted in the figure that shows the boundary of patterns. By looking at the highlighted scenarios, I can see that they are associated with the tasks 3, 7, 9, 11, and 12. In particular, changing the expertise level of task 12 to the novice (where all the other task is executed using an expert designer) has yielded the highest process duration and at the same time, the minimum variance in 1000 simulation results. On the contrary, changing the expertise level of task 7 to an expert (where all the other tasks are executing using a novice designer) has resulted in the minimum yet the widest process duration chart.

The result of scenario analysis in this section implies that finding the best (the most appropriate) task execution policy to a large extent depends on finding the most feasible compromise between process duration and quality. Using experts in running tasks while can significantly reduce the unnecessary iterations; it might result in increasing the total cost and duration of DP so far. However, the results obtained at this step is rich enough so that can support managers and decision-makers in finding the most appropriate process plan (including process route and task execution).

4.6. Summary: further discussion on the simulation results

This section was dealt with examining the functionality of the CAM software from the perspective of process modelling. In particular, I investigated the impact of modelling and simulation on process planning. Three different versions of the Signposting system used in this regard.

Rebuilding the simulation models and running them in the ASM toolbox revealed that identifying the most appropriate process plan is a non-trivial task which depends on a number of elements both at the task and at the process levels. Some of them that addressed in this section are probabilistic distribution of tasks properties (such as duration and cost), measuring aspects of iteration (or success condition), the knowledge of human resource on a given task (novice or expert), and the mechanism of precedence and dependency between tasks (due to direct interaction, or through their parameters, or through their actors).

These elements can be termed as the basic requirements of a modelling effort, as the result. For example, our study compared three models that in two of them (the OSM and the ESM), tasks are connected together through sharing the information on the same parameters. In ASM, on the contrary, the mechanism of information flows (i.e., iteration flows) are of central to the overall performance of the model. In this study, I looked at these issues and their effect on process planning issues (such as sequencing and scheduling) from a comparative point of view.

Obviously, the study presented in this section is not about to distinguish different versions of Signposting models in order to say which is good and which is better. In Fact, it has been more about the functionality of CAM (and the ASM toolbox, in particular) in modelling and simulating different types of modelling, i.e., to show how a DP can be modelled in a variety of ways.

Therefore, as far as related to the functionality of CAM, the study in this section shows that CAM is rich in capturing multiple sources of uncertainty and their effect on the simulation outcomes. For example, the pre-processing, processing, and post-processing parts in the ASM toolbox (see Section 2.3 for further information) provides a flexible environment for customising an existing function or adding a new function to the model.

Visualisation in CAM can support the user with a variety of tools (charts, diagrams, and plots) in an interactive way. The Excel plugin in CAM is a notable option, given the possibility of further analysis of the result of simulation in other platforms, such as what I did in making Gaussian Functions (see Figures 4.16 and 4.19). This can be seen as the additional functionality of the software when integrating with other platforms.

From a modelling perspective, further investigation can be suggested for future research. I already addressed issues related to identifying the best process plan, process route, and task execution policy. Any of these analyses can be discussed in more detail while finding the best iteration policy can be suggested to address, perhaps through changing the iteration flows. Our understanding represents that the ASM framework is more likely to show a higher sensitivity when changing the iteration flows, compared to the other Signposting versions.

Eventually, it should be noted that the quantitative study presented in this section (including re-building the models, simulation, and visualising them) totally took almost six months for the author to accomplish. Many individual discussions conducted during this time, to make the models exactly similar to what they originally proposed, and for getting feedback on the result of simulation and visualisation.

Chapter 5

5. Implications and conclusions

This chapter summarises and concludes the report. First, the five specific research questions introduced in Chapter 1 are revisited and the answers to them developed throughout the report are briefly summarised in Section 5.1. Then, the limitations of this research are discussed in Section 5.2; and finally, opportunities for further work are highlighted in Section 5.3, and the report is concluded in Section 5.4.

5.1. Key findings and research contributions

This section reviews the research contributions of this report and summarises the key findings under the answers to the five specific research questions stated in Chapter 1.

> **Specific RQ1:** What are the key characteristics of CAM software (what aspects of the software) that should be investigated?

This question addressed through dividing the process of developing a model into classes of model, building, simulation, and visualisation. In terms of model building, creating and modifying a new module in the software, compatibility between different toolboxes (to prevent starting the modelling from scratch), and preparations and tutorial raised an issue. During the model simulation, flexibility in generating multiple outputs (views), running and re-running a simulation, data analysis and optimisation algorithms (in case of a DSM model) and changing/switching between modelling views have been of significant impact. Regarding the visualisation, importing and exporting the data and integration of CAM with (popular) modelling packages have been two of the most important issues. All these accomplished through following the below steps:

- *Systematic literature review*. To answer to RQ1, the author reviewed the historical background and the key publications of CAM toolboxes. It was the first step to understand the state-of-the-art of research in aspects of Process, Dependency, and Change Modelling in Cambridge EDC, and to understand the underlying research in the group before developing the CAM software that resulted into development of CAM, referring back to 1997. Part of the extensive literature review is presented in Chapter 2.
- *The categorisation of the literature*. The next step to identify the key characteristics of CAM was to categorise the publications according to the main CAM toolboxes, ASM, DSM, and CPM. They are presented in sections 2.2, 2.3, and 2.4, respectively. Different toolboxes of CAM represent different sets of functions which have been improved over the years. Review of publications associated with each toolbox provided rather a good understanding of a range of functionality for each toolbox.
- *A general list of CAM functions come from the literature review*. A master list of functions associated with different CAM toolboxes is the result of extensive literature review together with studying CAM in-person. This list (part of which is presented in Tables 2-2, 2-3, and 2-4) used then during the qualitative and quantitative analysis.
- *A specific list of CAM functions come from the interviews*. The key characteristics of CAM are the set of questions that used in a questionnaire during the qualitative analysis in Chapter 3. It included such concerns on the software interface, compatibility of CAM with other software packages, and some detailed issues related to the model building, simulation, and visualisation of the results. For this purpose, as is shown in Fig. 3-1, the author put the outcome of literature review, face-to-face interview with internal users (the EDC Colleagues) and the external E-mails together to understand those aspects of CAM functionality that realised to be more important for the (internal and external) users.

Specific RQ2: What aspects of CAM software have been more popular between internal and external users, with respect to the different contexts such as academia and industry?

In addressing this question, several aspects of the user were taken into account when designing and releasing the questionnaire: educational level, educational background, and the affiliation (academia and/or industry). The results of survey presented in Chapter 3 show that majority of the respondents have been from academia and used the CAM for the purpose of research. The results also show the relative satisfaction of the users both in terms of the overall functionality of different toolboxes and in terms of overall CAM support. The parallel coordinates illustrated in Sec. 3.5.2 for instance focused on the view of the industrial users (see Fig. 3.3). On the opposite side, the users enumerated a number of issues regarding the data analysis in DSM for example or building a new model in ASM. Further analysis of responses presented in Sec. 3.5.1 provided a detail investigation of the correlation between responses and the questions. Steps to perform the above-mentioned activities are summarised in the following:

- *Qualitative analysis of internal CAM users*. The complete answer to RQ2 can be found in Chapter 3, in which the author developed a qualitative analysis of CAM from the perspectives of internal and external users. From the internal side, a range of face-to-face interviews conducted with the EDC Colleagues, each one has been dealing with a specific aspect of CAM such as process and/or change modelling. According to the results presented in Section 3.3 (particularly, Table 3-1), they have mainly been concerned with compatibility issues such as the extension to generate MDM-based CPM, changing views between toolboxes, and the general interface.
- *Qualitative analysis of external CAM users*. To answer the RQ2 from the external users' perspective, the author utilised the information from analysis of individual e-mails, face-to-face interviews and the relevant literature review to develop a questionnaire. The target audience of the questionnaire concurrently selected from the analysis of users in IT system. Some of the highlights are presented in Section 3.4.2 and are followed by an extensive statistical analysis in Section 3.5. The results imply the DSM toolbox as the most frequent and most popular (in terms of satisfying the users' expectations) toolbox, the ASM as the most sophisticated one (in terms of model building and simulation), and the CPM toolbox as the most flexible one (in terms of simulation and visualisation

and compatibility). At a more general level, they addressed such data analysis issues in all toolboxes.

Specific RQ3: How the feedback received from interviews and surveys can be used to improve functionality of future CAM updates, through analysing the data?

It was exactly the main contribution of the whole Chapter 3: the qualitative analysis. In addressing this question, the author attempted to gather, classify, and criticise the outcome of interviews and survey. In particular, this is accomplished through:

- *Statistical analysis of correlations and contexts*. Answering the RQ3 primarily required a more detailed analysis of results that are accomplished in Section 3.5 using Statistical Inference (i.e., factor analysis) and Parallel Coordinates diagram. In fact, the feedback received from the interviews is going to be addressed through the undergoing research projects in Cambridge EDC.
- *Using data analytics for future software improvements*. From the survey side, this work supported the CAM support team (for future updates of the software) by providing a range of hints, insights, and patterns (from the Parallel Coordinates) to understand what aspects of CAM have been the focus of its users. Some of them are presented in Fig. 3-3. Further interpretations of the results remained for the internal group meetings. As the first-ever separate research being conducted on the functionality of CAM, the present work attempted to gather as much data as possible and classify them in a way that might be helpful for future improvements of the software.

Specific RQ4: what are the basic requirements of an advanced modelling software, for building, simulating, and visualising the outcomes in a friendly manner?

It is very difficult to develop a user-friendly software that can properly reflect the iterative, uncertain, and dynamic nature of design projects and be accessible like CAM free of charge. In this report, aspects of information dependency between design tasks, the level of quality in performance-related parameters, and the duration uncertainty of design project discussed and compared. In general terms, it is concluded that a modelling software should be first of all be applicable to a variety of problems (project-independent). It should be flexible enough so that can capture the same design process at different levels of abstraction. At a more detailed level, a modelling tool should be able to capture the time, cost, and performance uncertainty of design, while providing a range of views to support a variety of decision-makers. The assumption is that each modelling view emphasises on certain information while omitting others (Browning, 2010). It should also be sensitive to change in the input information: an example of scenario analysis on finding the best task execution policy provided in Sec. 4.5.3 (see the Table 4.4 and Fig. 4.19). The supportive steps in the following determines the pathway to the above findings:

- *A comparative study of quantitative analysis of CAM*. CAM is substantially a research-based software that has been proposed from previous research for future research. Therefore, its functionality largely depends on the appropriate interpretation of feedback from users. For adding a new function to the software, it should be applied and tested in a range of problems. That is sometimes challenging since each research problem has its own settings (belonging to a specific context) and might require a different set of functions. As the result, to answer the RQ4, it was not enough to rely on the feedback obtained from users, and there should be a mechanism to investigate how software functions work in different settings in a similar environment.
- *Re-architecting Signposting Models*. To answer RQ4, by addressing the above challenge, the author relied on the widely used *Signposting* models that have been the focus of many studies in the past two decades. Doing this way, the original version of the model and two of its major extensions selected for a comparative study. The idea was to re-simulate the models, keeping their fundamental ideas, in the ASM toolbox of CAM. The whole Section 4 of this report is dedicated to re-building, simulating, and visualising these models in CAM. The comparative determined that different versions of Signposting can be distinguished according to their architecture as *information*-driven (OSM and ESM) and *dependency*-driven (ASM) process modelling tools. The simulation outcomes provided a wide range of insights, not only on the functionality of existing icons in CAM but also acting as a baseline for future Signposting improvements.
- *Sensitivity analysis for process optimisation*. Due to the capability of CAM in making discrete-event simulation models supported by a sort of what-if questions, the present work

moved beyond the simulation results and applied a set of sensitivity analysis (presented in Section 4.4 and 4.5) to test the utility of CAM visualisation tools in practice. For this purpose, the author partially used the Excel charts to map out the data come from CAM as a CSV file. According to the results, the probabilistic Gantt and Parallel Coordinate diagrams have been of great help in mapping the process planning issues (such as the best process route or the best task execution policy). In addition, compatibility of the software with other software packages such as Excel gives a lot of opportunities to use them jointly.

Specific RQ5: What are the key process modelling characteristics that should be reflected in future modelling attempts?

Depending on the focus of modelling, there might be different purposes such as documentation, analysis, visualisation, and planning of design processes. Accordingly, they require a specific set of characteristics during the model building and simulation, such as those pertaining to the modelling of duration, cost, risk, information quality, resource, and concurrency. From a functional point of view, our findings confirm the functionality of CAM in modelling complex physical models of design processes. However, investigating the organisational side of the problem requires further studies and did not address in this report. Additional suggestion is presented in the following:

- *Function-based analysis of Signposting models*. Routed in the RQ1, process modelling characteristics are seen in this report from the interdependent perspectives of future CAM updates and future modelling attempts. I addressed them in Chapter 2 when reviewing the main CAM toolboxes and then in Chapter 4 during the simulation and sensitivity analysis. I addressed these issues in this report through comparing different versions of Signposting that have been developed for different specific purposes and hence required different sets of functions.
- *Statistical analysis of qualitative data*. In addition to the quantitative analysis of modelling functions, the author applied a deeper investigation of data obtained from the questionnaire. The outcome appeared in Figure 7-1 in the Appendix and discussed in Chapter 3.5. The premise was that each user is concerned with a specific aspect of CAM,

such as dependency modelling or change modelling, and consequently is dealt with a particular set of modelling functions. Therefore, a more detailed investigation of their feedback might result in a better understanding of their modelling requirements, thus helping us to get a better understanding of the key modelling characteristics in general.

5.2. Research limitations

This report provided a systematic methodology to investigate the functionality of CAM software qualitatively (through interview and questionnaire) and quantitatively (through simulation and analysis). A few methodological issues therefore should be acknowledged as the research limitations; they might be only related to the qualitative side of the report, the quantitative side, or might be common throughout the research:

As far as related to the qualitative analysis of CAM, through interview and survey:

1. *The rate of response from the external users*. It was much lower than the expectations of the team, 53 responses out the total 533 e--mails sent out (around 10%). Several reasons can be figured out as the result such as the time of release (it was right after the New Year holidays), the mechanism of survey (the questions basically came from the experience of internal CAM users), or the composition of users (they selected based on the number of downloads over continuous years). As the result, the generality of the research results cannot be completely proven. However, a range of discussions and interviews with academic experts and internal users provided to support the findings. In addition, the quantitative analysis of CAM designed after that to support the outcome of the qualitative assessments.
2. *Unavoidable subjectivity and inaccuracy of survey results*. It can be termed as an inherent part of any qualitative analysis. The proposed set of questions (used during the interviews and survey) were mainly come from the literature of CAM toolboxes and complemented through a range of discussions with the internal users. Further subjectivity might be related to the range of users when filling the questionnaire. There is no certainty that every two individuals have the same opinion on a given question (relating to a specific function of CAM). In addition, the low rate of response that received from the external users had some effect on the result of statistical analysis.

3. *Time-consuming implementation and evaluation of the method*. The qualitative analysis applied in this study contained both interviews and the survey. Concerning the former interview, it was not so straightforward to arrange the meetings with EDC colleagues (who had experience of using CAM at least for a year). As far as related to the survey, upon the literature review, the author attempted to run the interviews and discussion with CAM experts in parallel with designing of a survey. The questionnaire should be generic enough so that can cover all key aspects of CAM while being rigorous enough so that can provide a detail information on the functionality of each of the CAM toolboxes. Taking these issues into consideration, it took three months to get back at least 10% of the released forms. All these together, the author spent around six months working on the preparations, release, and analysis of the survey.

As far as related to the quantitative analysis of CAM, through simulation and analysis:

4. *Industrial implementation and evaluation of the developed method*. The comparative simulation of Signposting models presented in this report was based on an old case study that has been broadly used in the previous studies in EDC. In fact, the validity of results discussed with the report supervisors and a few experts in the fields. While this could satisfy objectives of the report in terms of addressing the functionality of CAM, the method was not demonstrated in the industry and the author did not have a chance to discuss the results with practitioners herself. However, the ASM toolbox that used for the comparative simulation in this report is a tool that has been designed for addressing the industry challenges and according to the industry requirements.
5. *Re-simulating an old wine in a new bottle*. Re-building a model that developed almost 20 years ago in a completely different simulation platform might look like bringing an old wine in a new bottle, but in reality, it was more challenging than it was initially looked like to be. Except the Applied Signposting Model (Wynn, 2007), the two other models of Signposting nee the Original Signposting (Clarkson and Hamilton, 2000) and the Extended Signposting (O'Donovan, 2004) were developed in different modelling and simulation environments. As the consequence, there were some functions that did not properly work in all simulation models and facing many design-time and run-time errors as the result. From a developer's perspective, these issues might be seen as improvement points in future updates of the software and that was exactly what the CAM support team was looking forward.

5.3. Opportunities for further research

Some of the research limitations might be seen as opportunities for future research, including:

- *Exploring the generality of findings*. This can be nevertheless true for both qualitative (come from the interviews and survey) and quantitative (come from the simulation) findings. Concerning the qualitative findings, conducting a more generic survey is an opportunity to get a deeper insight from internal and external users of CAM. To this end, this work suggests providing a systematic mechanism for receiving feedback from CAM users so that can be used for future assessments. However, as the starting point, this report can pave a way for seeing how users with different properties look at CAM.
- *Method validation through more industrial case studies*. An industrial evaluation of the results or even an additional case study would add significant value to research on Signposting while facilitating the development of enhanced methods. According to the findings of this report, multiple versions of Signposting represented rather the same behaviour in addressing high-level process characteristics such as scheduling and resource planning. However, the inherent uncertainty in predicting design process behaviour requires a more robust evaluation of the validity of results derived from the simulation.
- *Sensitivity analysis of simulation results*. In Chapter 4 of this report, the author ran a set of scenario analysis to compare the Signposting versions in addressing a few process planning issues, such as the optimal process plan, process route, or task execution policy. A wider range of sensitivity analysis such as relating to modelling iterations and confidence analysis seem relevant for further research. In fact, at the current version of ASM toolbox, the full classification of different modes of iteration (Wynn and Eckert, 2017) is not clear. Related opportunities for further research include investigation of how to map the dependencies between quality parameters or map multiple classes of dependencies between design tasks in ASM.
- *Taking other design domains into modelling consideration*. The focus of this work has been mainly on the process domain, where it is assuming that the design process is a sequence of tasks need to be performed at the certain level of confidence in their quality parameters. It is also assumed that each can be accomplished using a novice or expert designer. However, when it comes to the real-life implementation of such processes,

the process is affected by other domains such as the product properties (as the result of customer requirements) and the organisational properties (as the result of designers' communication and interaction, for instance). At the current version of ASM toolbox, consideration of socio-technical characteristics is not well defined. The belief is that modelling task-actor dependencies can potentially uncover many limitations of CAM and be used in future software updates or future modelling research.

- *Comparing functionality of CAM with other modelling packages.* As mentioned before, CAM is free of charge for research purposes and there is no hesitate that there is not a lot of software like CAM that can support a broad range of functions (including process, dependency, and change modelling) in an integrated platform. To get a better understanding of functions that should be and functions that can be included in the software, it is worth looking at different software platforms comparatively.

5.4. Concluding remarks

The functionality of CAM software is important in designing high-performance process models. This report presented a systematic methodology to investigate the performance of CAM in satisfying its broad users across the globe. The proposed approach involved almost all internal users of the software in Cambridge EDC along with a group of external users including industrialists. To complement the qualitative research, a comparative study conducted by refining three major extensions of the well-known Signposting systems. They were implemented in the ASM toolbox. Based on a comparative analysis of simulation results, the report contributed a benchmarking approach to help identify the characteristics that should be improved in the software and the characteristics that should be included in future modelling attempts. The outcome of this report partially shared with the CAM support to be considered in future software updates.

Appendix

Table 7.1. An overview of the questions and response options in the online survey

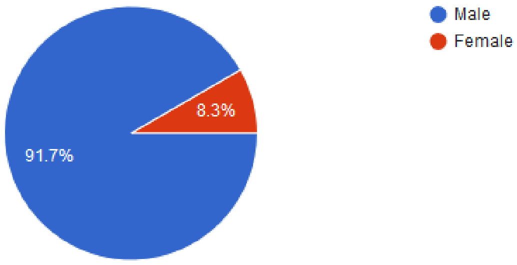
Number	Title of question	Response options
Section One. Personal information		
1	Gender	Male Female
2*	Age	Less than 25 Between 25 and 35 Between 35 and 50 Over 50
3	Educational degree	Bachelor (In-education or Degree obtained) Master (In-education or Degree obtained) Doctorate (In-education or Degree obtained)
4*	Educational background (in any major)	Engineering Business and management Computer and IT Science
5*	Employment status (at the time of using CAM)	Academic institution Industrial or manufacturing company Consulting company Other
Section Two. Previous experience in using CAM		
6*	For how long you have used CAM?	Less than a year Between 1 and 3 years Between 3 and 5 years More than 5 years
7*	What was your main objective of using CAM?	Personal advancement Student or research project Company uses Other
8	What was the current version of CAM being used?	2010 2012 2014 I do not remember

9	What was the Operation System that CAM has been installed on?	Windows Macintosh Linux
10	How would you rate the main toolboxes that have been used?	Design Structure Matrix (DSM) Applied Signposting Model (ASM) Change Propagation Method (CPM)
11*	By taking different aspects into consideration, how would you rate your overall experience in using CAM?	Not satisfied Needs improvement satisfied Very satisfied
Section Three. Towards further improvement of CAM		
12*	Which of the following training materials would your first experience of using CAM better?	Tutorial Video clips Workshops Other
13	What would be your main concern whilst installing the software?	Meeting the pre-requisites for the target platform Installing the software onto the target platform Configure the software following installation Verify the installation for use
14*	What aspect of CAM should be prioritized for improvement?	Interface and ease of use Functionality and performance Diversity of toolboxes Other
15*	Which function should be the focus of further improvement?	Building a new model Running a simulation Making outputs and reports Other
16	If you are using DSM or CPM, what would be your most important concern?	Compatibility and integration with other software Data analysis (e.g., partitioning, clustering) Flexibility in generating output files Extension and further plugins
17	If you are using ASM, what would be your most important concern?	Building a model (e.g., generate/modify a function) Running a simulation Organization of documentation of results Compatibility or import/export issues
18	In general, how would you rate the support of the software?	Inadequate Poor Good Excellent

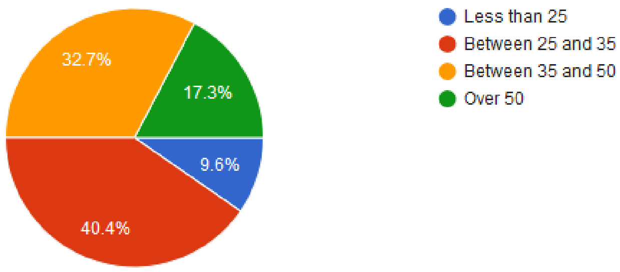
** Mandatory questions to answer*

(a) Section 1: Personal information

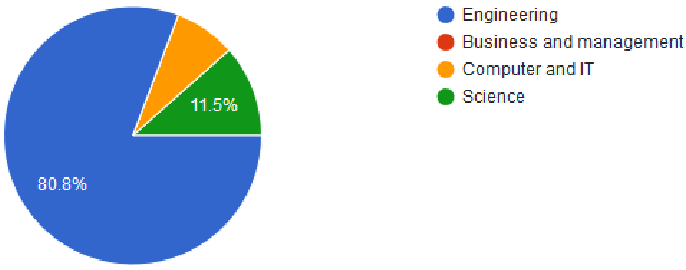
- Gender



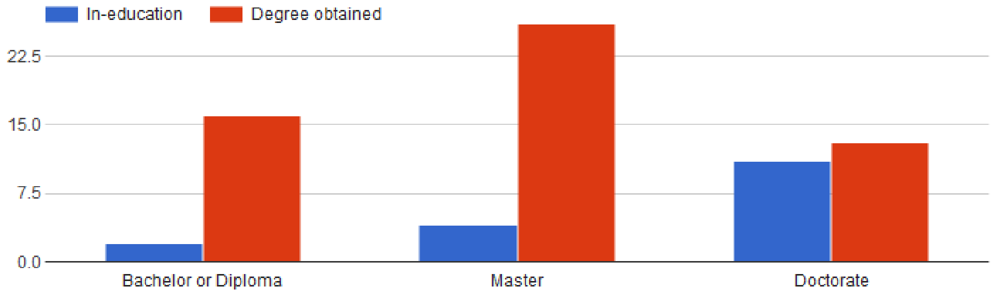
- Age



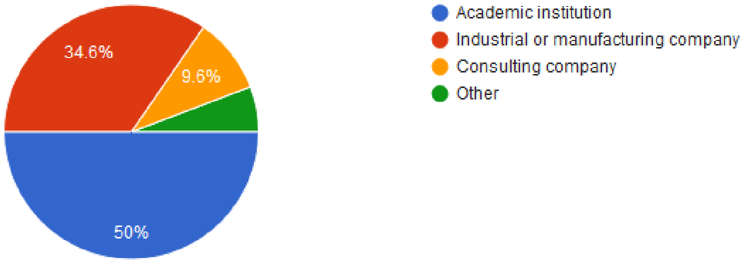
- Educational background (in any major)



- Educational degree

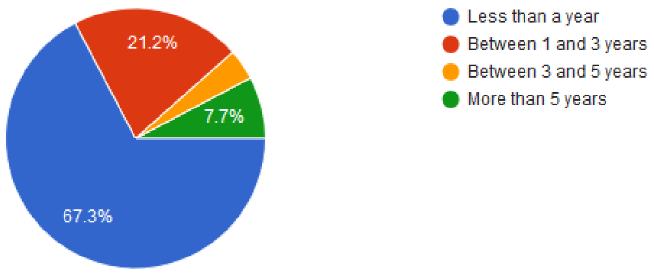


- Employment (at the time of using CAM)

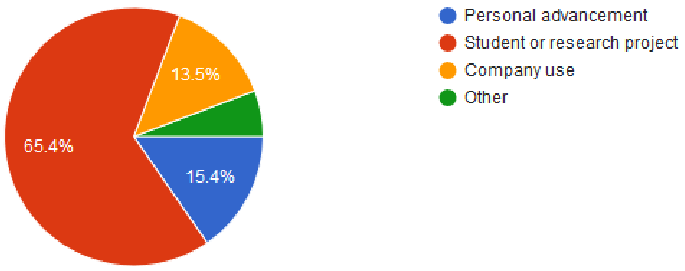


(b) Section 2: Previous experience in using CAM

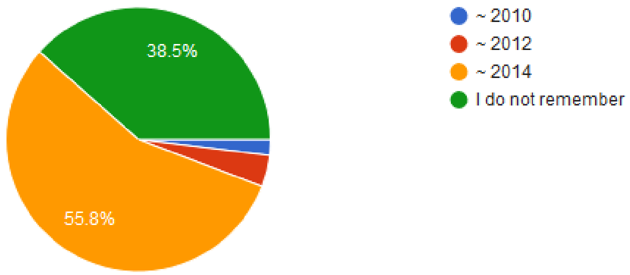
- For how long have you used CAM?



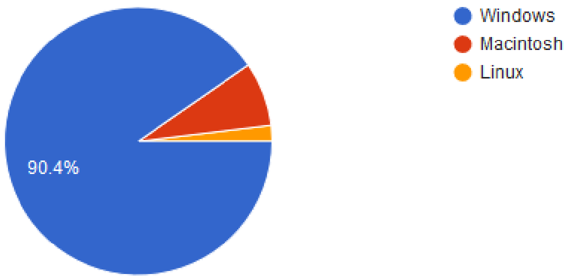
- What was your main objective of using CAM?



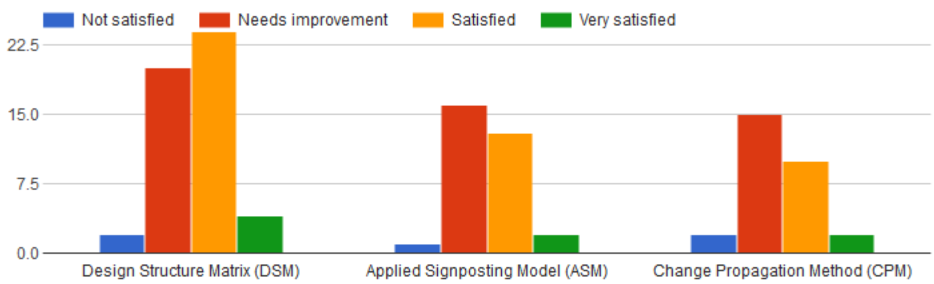
- What was the current version of CAM being used?



- What was the operation system that CAM has been installed on?

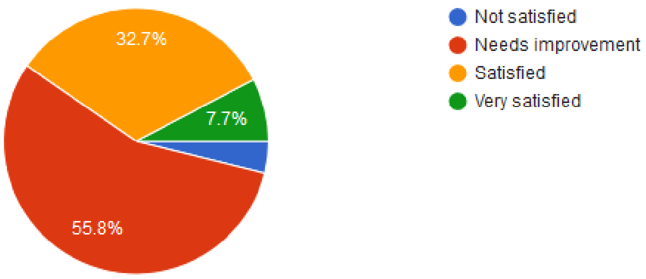


- How would you rate the main toolboxes that have been used?



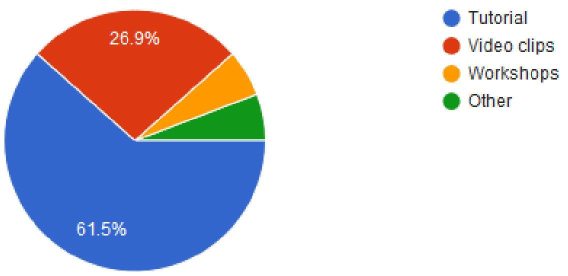
(b) Section 2: Previous experience in using CAM

- By taking different aspects into consideration, how would you rate your overall experience in using CAM?

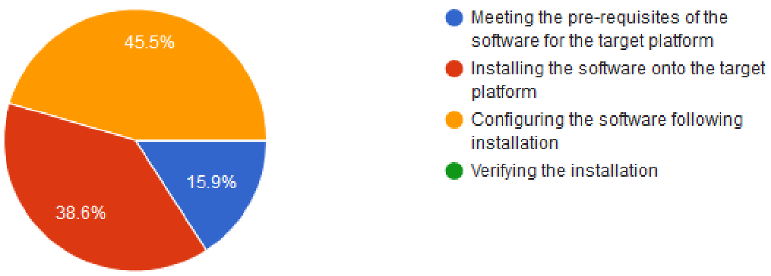


(c) Section 3: Towards further improvement of CAM

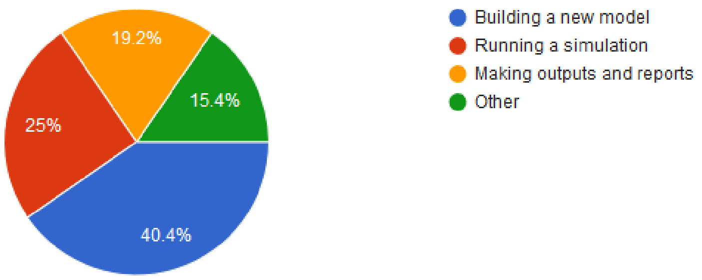
- Which of the following training material would make your first experience od using CAM better?



- What would be your main concern while installing the software?

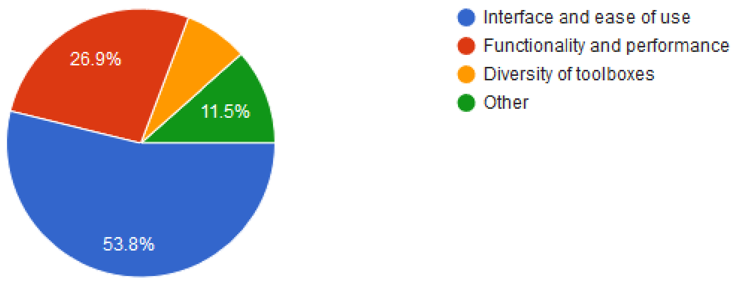


- Which function should be the focus of further improvement?

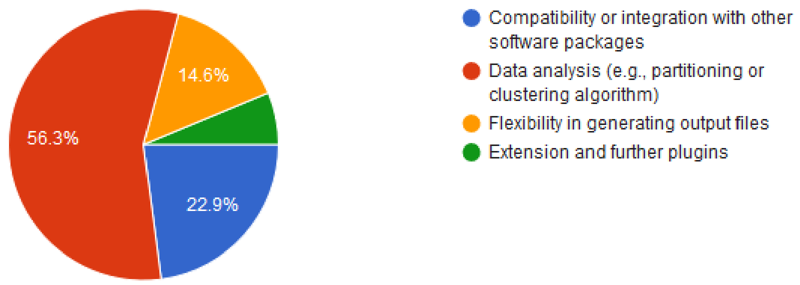


(c) Section 3: Towards further improvement of CAM

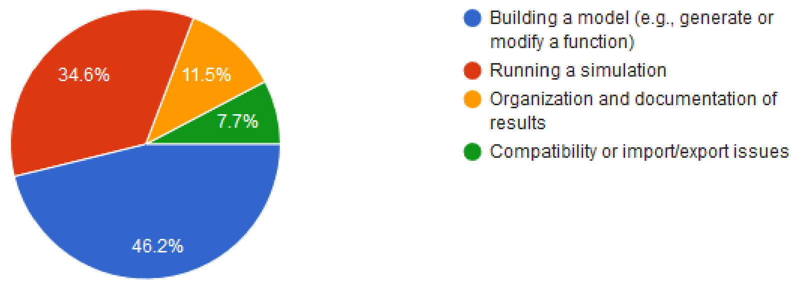
- What aspect of CAM should be prioritised for improvement?



- If you are using DSM or CPM, what would be your most important concern?



- If you are using ASM, what would be your most important concern?



- In general, how would you rate the support of the software?

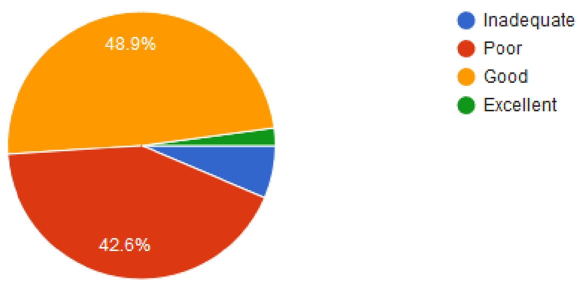


Figure 7.1. The primary result of online survey (based on 53 responses): (a) Section 1: Personal information, (b) Section 2: Previous experience in using CAM, and (c) Section 3: Towards further improvement of CAM

Table 7.2. The list of factors (questions) and their associated description in factor analysis (some questions in Table 7.1 are separated in multiple questions for the purpose of factor analysis)

Factor label	*Description*
Q1	Gender
Q2	Age
Q3	Educational degree: Bachelor or Diploma
Q4	Educational degree: Master
Q5	Educational degree: Doctorate
Q6	Educational background in any major
Q7	Employment status (at the time of using CAM)
Q8	For how long have you used CAM?
Q9	What was your main objective of using CAM?
Q10	What was the current version of CAM being used?
Q11	What was the Operation System that CAM has been installed on
Q12	How would you rate the main toolboxes that have been used: DSM
Q13	How would you rate the main toolboxes that have been used: ASM
Q14	How would you rate the main toolboxes that have been used: CPM
Q15	Overall, how would you rate your overall experience in using CAM?
Q16	Which of the following training materials would your first experience of using CAM better?
Q17	What would be your main concern whilst installing the software?
Q18	What aspect of CAM should be prioritized for improvement?
Q19	Which function should be the focus of further improvement?
Q20	If you are using DSM or CPM, what would be your most important concern?
Q21	If you are using ASM, what would be your most important concern?
Q22	In general, how would you rate the support of the software?
Q23	New version of CAM is coming very soon

Table 7.3. The outcome of factor analysis in SPSS: the table shows the first nine principal component that have the largest possible variance

Total Variance Matrix									
Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	3.540	15.392	15.392	3.540	15.392	15.392	2.528	10.993	10.993
2	2.334	10.147	25.539	2.334	10.147	25.539	2.116	9.201	20.194
3	2.036	8.850	34.389	2.036	8.850	34.389	2.027	8.812	29.006
4	1.885	8.197	42.586	1.885	8.197	42.586	2.000	8.698	37.703
5	1.692	7.355	49.941	1.692	7.355	49.941	1.781	7.744	45.447
6	1.544	6.715	56.656	1.544	6.715	56.656	1.581	6.873	52.320
7	1.337	5.815	62.471	1.337	5.815	62.471	1.562	6.792	59.112
8	1.225	5.328	67.798	1.225	5.328	67.798	1.544	6.715	65.827
9	1.060	4.607	72.405	1.060	4.607	72.405	1.513	6.578	72.405
10	0.893	3.883	76.288						
11	0.852	3.703	79.991						
12	0.771	3.350	83.341						
13	0.710	3.085	86.427						
14	0.547	2.378	88.804						
15	0.532	2.313	91.118						
16	0.417	1.814	92.932						
17	0.346	1.504	94.436						
18	0.319	1.386	95.821						
19	0.277	1.204	97.025						
20	0.236	1.027	98.053						
21	0.208	0.906	98.958						
22	0.162	0.703	99.661						
23	0.078	0.339	100.000						

Table 7.4. The rotated matrix of principle components (obtained from Table 7.3): 1st replication; the components that are located out of the clusters (Q5, in the first instance) will be removed from the analysis and the next replication will be run. The process will be continued until getting a sufficiently-correlated cluster of components.

Rotated Component Matrix									
	Principle Components (those with the largest variance obtained from Table 7.3)								
	1	2	3	4	5	6	7	8	9
Q14	.875	-.211	.024	.126	.017	-.044	-.168	-.064	.043
Q13	.848	-.160	-.167	-.044	.175	-.047	.064	.020	.013
Q8	.481	.459	-.161	.042	-.134	-.255	.272	.209	-.016
Q21	.386	.164	-.279	-.170	-.236	-.031	.338	-.149	-.319
Q2	-.167	.638	-.126	-.480	-.047	.115	-.125	-.043	.165
Q9	-.328	.630	.073	.121	-.054	-.105	.147	-.133	-.096
Q19	-.173	.607	.461	-.023	-.004	.009	.217	.048	-.109
Q1	-.052	-.479	.133	-.013	.019	-.320	.422	.168	-.438
Q16	-.006	.420	.313	.137	-.315	.338	.232	.089	-.310
Q4	-.083	.127	.826	-.107	-.084	.000	.111	.063	-.005
Q5	.064	.122	-.740	-.245	.043	-.280	.235	.147	.061
Q22	.239	-.015	.101	.739	.080	-.187	-.038	-.226	.034
Q23	.271	.059	.096	-.681	.030	-.371	-.049	-.009	.191
Q10	-.481	.172	-.055	.520	-.306	-.209	-.007	.052	-.070
Q12	.037	-.114	.054	-.055	.838	-.113	-.203	-.024	-.006
Q15	.173	-.002	-.243	.108	.725	.071	.287	-.048	-.170
Q7	-.049	-.061	.165	-.045	-.091	.780	.066	-.062	-.020
Q6	-.006	.335	-.023	-.009	.426	.545	-.032	.367	.060
Q18	-.040	.129	.027	.046	-.011	.091	.844	-.076	.076
Q17	.117	.055	-.094	.224	.086	.071	-.013	-.870	-.010
Q11	.262	-.075	-.225	.266	.073	.121	-.257	.593	-.321
Q20	.049	-.068	-.098	-.146	-.136	-.069	.038	-.101	.813
Q3	.045	.085	.363	.418	.116	.135	.213	.266	.472

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.
Rotation converged in 17 iterations.

Table 7.5. The final rotated matrix of principle components: 7th and last replication

	Rotated Component Matrix				
	Principle Components				
	1	2	3	4	5
Q14	.919	-.027	.169	-.107	.029
Q13	.892	.185	.026	.084	-.145
Q15	.104	.765	.190	.226	-.188
Q12	.136	.720	.091	-.462	-.020
Q6	-.064	.647	-.361	.049	.240
Q17	-.028	.021	.823	-.040	-.192
Q22	.210	.013	.728	.100	.236
Q18	-.222	.101	.134	.765	.117
Q8	.413	-.076	-.100	.666	-.031
Q3	.088	.157	.089	.125	.802
Q4	-.214	-.189	-.107	-.041	.687

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.
Rotation converged in 5 iterations.

Table 7.6. The list of dimensions (questions) whose responses used for multi-dimensional analysis in PCD

Dimension	*Description*
D1 (Q6)	Educational background in any major
D2 (Q7)	Employment status (at the time of using CAM)
D3 (Q8)	For how long have you used CAM?
D4 (Q12)	How would you rate the main toolboxes that have been used: DSM
D5 (Q13)	How would you rate the main toolboxes that have been used: ASM
D6 (Q14)	How would you rate the main toolboxes that have been used: CPM
D7 (Q18)	What aspect of CAM should be prioritized for improvement?
D8 (Q19)	Which function should be the focus of further improvement?
D9 (Q20)	If you are using DSM or CPM, what would be your most important concern?
D10 (Q21)	If you are using ASM, what would be your most important concern?

Bibliography

Ahmad N, Wynn D and Clarkson P (2013) Change impact on a product and its redesign process: A tool for knowledge capture and reuse. *Research in Engineering Design* 24(3): 219–244.

Aurisicchio M and Bracewell R (2013) Capturing an integrated design information space. *Journal of Engineering Design* 24(6): 397–428.

Browning T (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering Management* 48(3): 292–306.

Browning T (2010) On the alignment of the purposes and views of process models in project management. *Journal of Operations Management* 28: 316–332.

Browning T (2016) Design Structure Matrix Extensions and Innovations : A Survey and New Opportunities. *IEEE Transactions on Engineering Management* 63(1): 27–52.

Browning T and Ramasesh R (2007) A Survey of Activity Network-Based Process Models for Managing Product Development Projects. *Production and Operations Management* 16(2): 217–240.

Browning T, Fricke E and Negele H (2006) Key concepts in modeling product development processes. *Systems Engineering* 9(2): 104–128.

Chalupnik M, Wynn D, Eckert C, et al. (2007) Understanding design process robustness: A modelling approach. In: *ICED'07*, Paris, pp. 1–12.

Chalupnik M, Wynn D, Eckert C, et al. (2008) Investigating Design Process Performance Under Uncertainty. In: *TMCE08*, Kusadasi, pp. 1–14.

Chen H, Moullec M-L, Ball N, et al. (2016) Improving Design Resource Management Using Bayesian Network Embedded in Task Network Method. In: *ASME 2016 IDETC/CIE*, Charlotte, NC, pp. 1–15.

Clarkson P and Hamilton J (2000) 'Signposting', A Parameter-driven Task-based Model of the Design Process. *Research in Engineering Design* 12: 18–38.

Clarkson P, Connor A and Melo A (1999) Signposting, A Dynamic Approach to Design

Process Management. In: *16th International Joint Conference on Artificial Intelligence (IJCAI-99) Workshop on Workflow Management*, Stockholm.

Clarkson P, Melo A and Eckert C (2000) Visualization of routes in design process planning. In: *IEEE Conference on Information Visualization - An International Conference on Computer Visualization and Graphics*, IEEE Comput. Soc, pp. 155–164.

Clarkson P, Simons C and Eckert C (2001) Predicting Change Propagation in Complex Design. In: *ASME 2001 DETC/DTM*, Pittsburg, Pennsylvania, p. 081009.

Clarkson P, Melo A and Eckert C (2001) Visualization techniques to assist design process planning. In: *ICED'01*, Glasgow.

Clarkson P, Simons C and Eckert C (2004) Predicting Change Propagation in Complex Design. *Journal of Mechanical Design* 126(5): 788–797.

Eckert C, Zanker W and Clarkson P (2001) Aspects of a better understanding of changes. In: *ICED'01*, Glasgow.

Eckert C, Wynn D, Maier J, et al. (2017) On the integration of product and process models in engineering design. *Design Science* 3(3).

Eger T, Eckert C and Clarkson P (2003) Towards a change process planning tool. In: *ICED'03*, pp. 1–10.

Eisenbart B, Gericke K and Blessing L (2013) An analysis of functional modeling approaches across disciplines. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 27(03): 281–289.

Field A (2009) *Discovering Statistics Using SPSS*. *SAGE Publications*.

Gericke K and Blessing L (2012) An analysis of design process models across disciplines. In: *DESIGN2012*, Dubrovnik, pp. 171–180.

Gerwin D and Barrowman N (2002) An Evaluation of Research on Integrated Product Development. *Management Science* 48(7): 938–953.

Hamilton, JR (1999) The capture and representation of knowledge to support adaptive aerospace design. *PhD Thesis*, University of Cambridge.

Hamraz B, Caldwell N and Clarkson P (2012) A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design. *Journal of Mechanical Design* 134(10): 1–14.

Hamraz B, Caldwell N and Clarkson P (2013) A Holistic Categorization Framework for Literature on Engineering Change Management. *Systems Engineering* 16(4): 473–505.

Hamraz B, Caldwell N, Ridgman T, et al. (2015) FBS Linkage ontology and technique to support engineering change management. *Research in Engineering Design* 26(1): 3–35.

Hassannezhad M (2015) Model-Based Support for Management of Engineering Design Processes. Politecnico di Torino.

Hassannezhad M, Cantamessa M and Montagna F (2015) Actor-Based Signposting: A modeling tool to improve the socio-technical design processes. In: *ICED'15*, Milan, Italy, pp. 1–10.

Hassannezhad M, Cassidy S and Clarkson P (2017) Dynamic modelling of relationships in complex service design systems. In: *ICED'17*.

Heisig P, Caldwell NHM and Clarkson PJ (2014) Core information categories for engineering design - contrasting empirical studies with a review of integrated models. *Journal of Engineering Design* 25(1–3): 88–124.

Jarrett, J. P. (2001). Technology or methodology?: an approach to designing better turbomachinery. *PhD Thesis*, University of Cambridge.

Jarratt T, Eckert C, Caldwell N, et al. (2011) Engineering change: An overview and perspective on the literature. *Research in Engineering Design* 22: 103–124.

Keller R, Flanagan T, Eckert C, et al. (2006) Two sides of the story: Visualising Products and Processes in Engineering Design. In: *10th International Conference on Information Visualisation (IV'06)*, Ieee, pp. 362–367.

Keller R, Eckert C and Clarkson P (2009) Using an engineering change methodology to support conceptual design. *Journal of Engeering Design* 20(6): 571–587.

Kerley W, Wynn D, Eckert C, et al. (2011) Redesigning the design process through interactive simulation: A case study of life-cycle engineering in jet engine conceptual design. *International Journal of Services and Operations Management* 10(1): 30–51.

Koh E, Caldwell N and Clarkson P (2013) A technique to assess the changeability of complex engineering systems. *Journal of Engineering Design* 24(7): 477–498.

Krishnan V and Ulrich K (2001) Product Development Decisions : A Review of the Literature. *Management Science* 47(1): 1–21.

Le H, Wynn D and Clarkson P (2012) Impacts of concurrency, iteration, design review, and problem complexity on design project lead time and error generation. *Concurrent Engineering: Research and Applications* 20(1): 1–13.

Little J (1970) Models and Managers: The Concept of a Decision Calculus. *Management*

Science 16(8): B466–B485.

Melo A (2002) A State-Action Model for Design Process Planning. *PhD Thesis*, University of Cambridge.

O'Donovan B (2004) Modelling and Simulation of Engineering Design Processes. University of Cambridge.

O'Donovan B, Clarkson P and Eckert C (2003) Signposting: modelling uncertainty in design processes. In: *ICED'03*, Stockholm, pp. 1–10.

O'Donovan B, Eckert C and Clarkson P (2004) Simulating Design Processes to Assist Design Process Planning. In: *ASME 2004 DETC/CIE*, Salt Lake City: ASME, pp. 503–512.

Ramdas K (2003) Managing product variety: An integrative review and research directions. *Production and Operations Management* 12(1): 79–101.

Shapiro D (2016) Modelling and Analysing Changes in Evolutionary Design Processes. *PhD Thesis*. University of Cambridge.

Shapiro D, Curren M and Clarkson P (2016) DPCM: a method for modelling and analysing design process changes based on the Applied Signposting Model. *Journal of Engineering Design* 27(11): 785–816.

Smith R and Eppinger S (1997) A Predictive Model of Sequential Iteration in Engineering Design. *Management Science* 43(8): 1104–1120.

Stacey M, Clarkson P and Eckert C (2000) Signposting: An AI approach to supporting human decision making in design. In: *ASME 2000 DETC/CIE*, pp. 1–10.

Stevens C (2012) *Topic Maps for the Cambridge Advanced Modeller Technical abstract*. *Fourth-year Undergraduate Project*.

Wyatt D, Wynn D, Jarrett J, et al. (2012) Supporting product architecture design using computational design synthesis with network structure constraints. *Research in Engineering Design* 23(1): 17–52.

Wynn D (2007) Model-Based Approaches to Support Process Improvement in Complex Product Development. Cambridge: Engineering Design Centre.

Wynn D (2010) *Modelling and simulating a product development process using the Applied Signposting Model in the Cambridge Advanced Modeller*.

Wynn D and Clarkson P (2005) Models of designing. In: *Design Process Improvement: A Review of Current Practice*, Springer Berlin Heidelberg, pp. 34–59.

Wynn D and Clarkson P (2018) Process models in design and development. *Research in Engineering Design*, Springer London 29: 161–202.

Wynn D and Eckert C (2017) Perspectives on iteration in design and development. *Research in Engineering Design*, Springer London.

Wynn D, Clarkson P and Eckert C (2005) A model-based approach to improve planning practice in collaborative aerospace design. In: *ASME 2005 IDETC/CIE*, California, pp. 1–12.

Wynn D, Eckert C and Clarkson P (2006) Applied Signposting: A modeling framework to support design process improvement. In: *ASME 2006 IDETC/CIE*, Philadelphia, pp. 1–10.

Wynn D, Nair S and Clarkson P (2009) The P3 platform: An approach and software system for developing diagrammatic model-based methods in design research. In: *ICED'09*, Stanford.

Wynn D, Wyatt D, Nair S, et al. (2010) An Introduction to the Cambridge Advanced Modeller. In: *Proceedings of the 1st International Conference on Modelling and Management of Engineering Processes (MMEP 2010)*, pp. 19–20.

Wynn D, Caldwell N and Clarkson P (2010) Can change prediction help prioritise redesign work in future engineering systems? In: *DESIGN2010*, Dubrovnik.

Wynn D, Grebici K and Clarkson P (2011) Modelling the evolution of uncertainty levels during design. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 5(3): 187–202.

Wynn D, Cassidy S and Clarkson P (2012) Design of robust service operations using cybernetic principles and simulation. *DESIGN2012* DS 70: 331–342.