

An Introduction to the Cambridge Advanced Modeller

David C. Wynn, David F. Wyatt, Seena M.T. Nair and P. John Clarkson

1.1 Introduction

Complex products and their development processes may be viewed as systems, whose different aspects can be modelled as networks of interactions between elements in different domains. Many approaches have been proposed to explore, support or improve engineering processes by building such models. Developing these approaches, and applying them to problems of realistic complexity, often requires specialised computer software suitable for manipulating large data sets. However, creating suitable tools can be difficult—because software development is time-consuming and requires skills that many researchers and practitioners do not possess.

We developed an approach which aims to address this problem by recognising the iterative nature of modelling research and its often tight coupling with prototype software development, and by reducing the effort of software prototyping and revision within this process. The approach is enabled by, and embodied in, the Cambridge Advanced Modeller (CAM)—a configurable software platform we have developed, refined and applied over several years and through a number of research projects.

The CAM, previously known as P3 Signposting, provides interchangeable diagram, DSM, and force-directed-layout views for constructing and visualising models of complex systems. It can be configured without programming to implement a wide range of modelling frameworks, as long as they can be specified declaratively in terms of the elements allowed in a model and the constraints upon how elements can be connected. Because it is based on configuration, rather than programming, CAM allows modelling frameworks to be rapidly refined while ensuring that the implementing tools remain stable and useable. CAM-based tools are relatively easy to extend or customise—since only knowledge of the configuration approach is required, and not knowledge of implementation code.

The CAM user interface is constructed automatically from this configuration to provide an experience tailored to the modelling approach at hand. In situations where models need to be analysed, e.g. through discrete-event simulation, the platform can be extended through Java plug-ins which appear as options and menu

buttons in the tool. Since the platform includes facilities for displaying the results of such analysis in graphical form, development and integration of analysis “toolboxes” often requires very little user interface development. The CAM platform has been used at the Cambridge Engineering Design Centre to assist a number of research projects involving development of systems modelling and analysis methods, providing an alternative to general-purpose diagramming tools or spreadsheets as a way to work with dependency data.

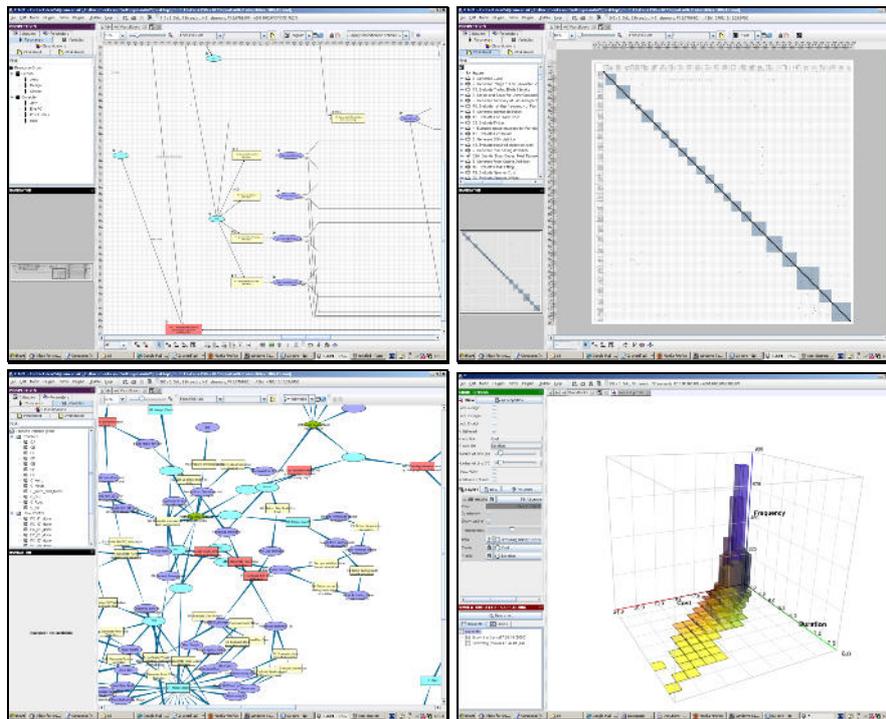


Figure 1.2. CAM provides functionality to visualise and manipulate models as diagrams, dependency matrices and force-directed layouts, as well as to explore model properties and analysis results.

Figure 1.3 shows the architecture of the CAM system and how it may be used to develop toolboxes for modelling (Toolbox 1), modelling supported by custom code (Toolbox 2), or modelling and computational analysis (Toolbox 3).

1.2 Platform functionality

CAM includes facilities for constructing and visualising linkage models using diagrams, DSMs or force-directed network layouts. Meta-models define the types of element that may be used in particular model types, the properties of those

elements, and the low-level “syntax” of the model in terms of the ways in which elements may be connected together. It is also possible to specify how the elements and their connections appear in different views of the model.

The diagram interface, similar to general-purpose diagramming software and shown in the top-left corner of Figure 1.2, allows the user to place diagram elements from a palette onto the worksheet, edit their properties, and connect them using arrows. An expanding grid system gives the user full control over layout, even for very large diagrams; elements may also be moved into sub-sheets, which can be interactively expanded to show their contents or collapsed to hide detail.

An alternative interface is the DSM view, shown in the top right corner of Figure 1.2. In a DSM, each element being displayed corresponds to a row and a column of the matrix, while connections between two elements are shown as marks in the requisite matrix cells. The ordering of the rows and columns can be manipulated to highlight structural characteristics of the model—either manually or algorithmically. Grouping of elements into nested clusters is also supported in the DSM.

Finally, for situations where a graphical layout of a model is desired but a manually-laid-out diagram is not feasible, a force-directed-layout view can be used. An example is shown in the bottom-left corner of Figure 1.2.

When using CAM to construct models which support computational analysis, the results of the analysis can be explored using charts and tables (an example is shown in the bottom-right corner of Figure 1.2). Parts of a resultset can be picked out and manipulated separately, and all results can be exported for further analysis in spreadsheet software (or other packages).

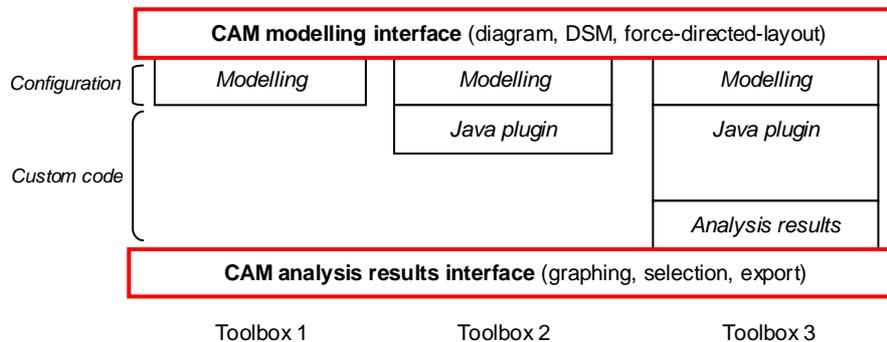


Figure 1.3. An overview of the architecture of CAM.

1.3 Built-in toolboxes

The CAM distribution includes implementations of various modelling languages, simulation methods and other analytical tools proposed in the literature of design process systems, including:

- System dynamics stock-and-flow simulation
- DSM manipulation algorithms (clustering, partitioning, banding etc)
- Applied Signposting modelling and discrete-event simulation

4 Wynn, Wyatt, Nair and Clarkson

- Change Prediction Modelling.

We hope that these built-in toolboxes will provide a starting point to assist the software prototyping of related research ideas.

1.4 Reflections and future developments

The CAM approach and software framework has been used within a number of research projects at the Cambridge EDC and beyond, in areas as diverse as design process modelling and simulation, requirements traceability modelling for change impact assessment, product architecture modelling, and product architecture synthesis. During the course of this work we have found that, by significantly easing the implementation of modelling languages, the CAM has allowed projects to produce usable modelling tools that can be tested in practice—aiding the creation of methods that are pragmatically feasible and relevant to real situations.

A number of issues for further development have also become apparent. Since the software tool is designed to be generic across modelling languages, it can be complicated to use in some situations. The ease of creating and modifying modelling languages has led to a need for functionality to control and track versions of the modelling languages as well as models themselves. Future developments will aim to address these issues, as well as allowing for more structured use of analysis methods to design and carry out "virtual experiments" on models. In addition, we recognise the plurality of software tools available and so we aim to improve interoperability between CAM-based tools and others.

1.5 Further information

Wynn DC, Nair SMT and Clarkson PJ (2009) The P3 Platform: An approach and software system for developing diagrammatic model-based methods in design research. In Proceedings of the 17th International Conference on Engineering Design, ICED'09, Stanford, CA.

Wynn DC (2007) Model-based approaches to support process improvement in complex product development. PhD thesis, University of Cambridge.