Defining Multi-Disciplinary Views of a Computer Model of Buildings

Robert W. Amor and Dr John G. Hosking
Department of Computer Science, University of Auckland, Auckland

ABSTRACT
The emergence of international standards for the representation of building product data will have a major impact on the development of design analysis tools in the building industry. Many foresee this impact as being an increase in the number of design tools which are poorly suited to the actual methods of the users. This will force the user to design and interact with the computer in a predetermined manner, a manner which is determined by the information requirements of the particular design tool.

One approach to alleviating this problem is the development, and maintenance, of multiple views of the building model as required by different disciplines in the building profession (eg. architects, engineers, developers, etc.). This solution will provide a flexible method of accessing information in product data models to facilitate use by different building professionals with varying specialities and levels of expertise.

1. Introduction

The emerging standard ISO 10303, colloquially known as STEP (STandard for the Exchange of Product model data), is being developed for the exchange of product data in the fields of manufacturing, architecture, engineering, construction and electronics. While it is likely to be many years before a full representation of buildings can be supported by this standard it is almost guaranteed success through the support of major commercial and governmental firms such as Boeing, the US Army and the US Navy.

The STEP standard will become the preferred method for transferring building information between various CAD systems, simulation tools and knowledge based systems. It will also make available technical information about products in a form that can be accessed by the design tools. This will help make these tools more attractive to use in the design of a building. Such design tools have been available to designers for many years, but there have been several obstacles to their acceptance and use. Amongst these are:

- The duplication of effort in describing the same building to multiple design tools. To gain information about varied aspects of the building design the same data must be input into each tool in its own specific format.
- The language used to describe a building to any tool. This is often arcane and in many cases unreadable to a human operator. This leads to long

learning curves, and many opportunities for errors in describing the building to the tool.
- The level of expertise required to describe a building to the design tools. A high level of expertise in the specific area, and a good knowledge of the physics and mechanical aspects of the components involved, is often assumed.

While the STEP standard will help overcome some of the problems detailed above, it will do little to change the interface to design tools, apart from provide a view of a building suitable for computer based transfer of building data.

In this paper we look at what is required to establish multiple views to a data model such as STEP. Section two details other work in the area. Section three looks at the ICAtect project which will form the base model for this research. Section four examines the requirements for multiple views. Section five provides a summary and future directions.

2. Previous work

The problem of interacting with users in a manner with which they feel comfortable has long been recognised. Only recently, however, has the emphasis of the conversation with a design tool shifted from requirements necessitated by the structure of the program and the language in which it was written, to something closer to the views of the user. Unfortunately, most of the tools which incorporate such user-friendly interfaces are aimed directly at one market in the industry, eg. architects, and so do not address the needs of other users. There are exceptions of course and we will describe two approaches taken by various research groups.

Views via model sub-sets

In this approach a view to a data model is provided by specifying a sub-set of the global model which will be visible in a particular view. This approach relies upon the central model containing all the objects and abstractions that a user would wish to use. This approach is easily amenable to current database technology as the mapping to a view is always one-to-one, and the database system can use its locking mechanisms to control access to the data and to maintain consistency of updates even when multiple users have overlapping views. The major projects using this approach are:

- At the Delft University of Technology [1] multiple views to a prefabricated concrete structural system are being explored. Based upon graph theory these views are sub-sets of the global model containing information that is applicable to a given user.
- The STEP standard [2] provides views, called application protocols, which are amalgamations of lower level objects from a resource model. These views are aimed at a particular industry (eg. piping, electrical wiring, structures) but when data is transferred to another user common information from the lower resource model is visible to that user.

Views via functional abstractions

In this approach a view to a data model is provided by specifying functional abstractions of the global model which are visible in a particular view. This approach relies upon the ability to define a mapping from the global model to a specific view and to enable data to be converted in both directions as needed. This makes the approach more powerful than the model sub-sets, and in fact subsumes that approach. In all the systems described below this entails specific code to handle mappings between objects in the various views, and in most cases specially tailored knowledge bases, or code attached to the global objects, to maintain the conceptual views for each user.

- At North Carolina State University [3] the Galileo-3 project researching concurrent engineering has a system for providing multiple user views. They introduce the concept of a field of view as a subset of the parameters available in a view. Using a hierarchy of these fields of view enables a user to expand or diminish the scope of information visible to them.
- At MIT [4] the DICE (Distributed and Integrated environment for Computer aided Engineering) project provides multiple levels of both functional and geometric abstractions of objects for views. They use the notion of functional spaces representing functional viewpoints of a physical space containing the primary representation of objects. The mappings and varying object behaviour under different views must be provided by the objects so a global object must know how to translate itself into the form required by any functional view.
- The IFE project [5] provides explicit procedures to handle different classes of user at various levels of expertise through different form sets to represent the same concept. The demonstration of this project provides an intelligent front end to the ESP simulation tool in which they examine how to provide a user model (of an engineer) through a predefined set of forms.

The field of database schema integration [6] provides insights to the problems in mapping between the various perspectives of similar objects when integrating databases from various organisations. They categorise the types of perspective mappings as: identical perspectives; different but equivalent perspectives; compatible specifications; and incompatible specification.

Figure 1. Structure of ICAtect

The field of visual programming and the MViews project at the University of Auckland [7] also provide useful insights. MViews provides multiple user defined views to an object oriented program, maintaining the integrity of the program, and change propagation, between the various views. Many aspects of software engineering mirror those in building design making the application of software engineering techniques a viable approach to solving building industry problems.

3. The ICAtect project

As a basis for this research we needed a model to form the base view of a building to provide user views to. As STEP is not in a form which is useable for such a project, we will use the model provided in the ICAtect system (see Figure 1. for the structure of ICAtect) [8, 9]. ICAtect has been developed over the last three years to examine a method of design tool integration, specifically for preliminary architectural design. The aim is to make quality information available to architects from these design tools as they examine variations in their building design to satisfy the design requirements.

The core of the current ICAtect system is a model of a building capable of holding all information required by a range of design tools useful to architects in the preliminary design stage. This common building model (CBM) was created from an analysis and amalgamation of the objects and attributes used by various design tools to describe a building for their simulation purposes.

To allow data to pass between ICAtect and the design tools a mechanism for moving common building data, mainly geometric, is necessary. This is achieved by providing a mapping of data between every design tool required and the CBM, enabling ICAtect to move its description of a building from one design tool to the next as needed. This is similar to the view methods provided by functional abstraction systems, as described in section 2.

To allow user interaction with ICAtect there is an interface that is structured to be easy and intuitive to use. It provides one language to describe a building to any design tool; and, through the use of constraints on objects and attributes in the CBM, validates the design as a consistent building design as information is entered.

ICAtect is structured to allow the user to analyse a building design at an early stage when very little of the building information has been specified. This is done by providing much of the detailed information required by the design tools as defaults, once the system knows the type of building being constructed and its general locality.

Analysis of ICAtect

Analysis of the prototype ICAtect system and work performed last year on a graphical interface to ICAtect [10] highlighted some of the deficiencies of the present model and system. These problems fall into two areas:

- Having constructed a CBM from various simulation tool models, access to the resulting system still requires the user to think and work at the same semantic level as the various simulation tools, ie. the user must still describe the building design in a language similar to those of simulation tools.
- The model has not been tested with applications outside the class of simulation tools. With the emergence of greater numbers of knowledge-based systems this omission needs to be dealt with.

These problems are closely related in that both are concerned with differing views of the CBM for different applications. The first problem can simply be addressed by tailoring the user interface to the language of an architect. However, viewing the problems as a single issue tends to suggest that we take a more global view to the problem. Indeed the amount of similar work undertaken in the areas of quantity surveying, thermal engineering and structural engineering [11, 1, 4, 5] demonstrates quite clearly that this project has much scope outside the field of preliminary architectural design for which it was first envisaged.

To address these problems we must generalise the structure of ICAtect's user interface subsystem to make it capable of handling multiple views of the building model. This structure will allow views of the database to be tailored not just for different classes of design tools, such as simulation and knowledge-based systems, but also for the different classes of practitioner in the building profession such as architects, engineers, developers, etc.

The intention is for ICAtect to provide different views of the building for different classes of users (see Figure 2. for revised structure). Thus, an architect has quite a different view of the building from a structural engineer. This ensures that users of each class are addressed in a language they understand, and are presented with the view of the building most relevant to them.

Figure 2. Multiple views to ICAtect

4. Multiple views

As a first step to structuring a solution for multiple views we must define what constitutes a view. There are several components to a view:

From the users viewpoint a view must be able to be personalised to:

- Provide the information required by the user in the form they request, request information that the user will know, only request information that is needed, and use other sources to find other required information, for example other users in the system, defaults for specific objects, or from external databases.
- Work at the same level of comprehension as the user, this involves tailoring concepts to the users level of comprehension so as not to overwhelm the user.
- Follow the methods of design the user is used to.

The provision of views of the form described above requires work in two major areas, one being the creation of mappings, the other to handle multiple users and the aspects of concurrency.

Providing mappings requires:

- A formal method for defining the mapping between the two models. These definitions should be able to be defined by the user in a friendly format.
- Management of the data transfer between the models via the defined mappings. Only one mapping should have to be defined and the system should be capable of mapping data in the required direction.
- Flexibility of the mapping. The user should be able to easily modify the views as they are working to suit the task at hand.

Providing for multiple users and concurrency requires:

- Ownership control to manage who owns the data, for example, who has the right to change the data, add data, delete data, etc.
- Change management to control the propagation of changes and to control who should see the effect of changes and the manner in which they are informed of the changes.
- Negotiation for conflicts between users requirements. This is partially handled by the concept of ownership control, but in the case where two or more users have rights to modify a data item, a method is required to obtain a compromise.
- Version management to handle alternative designs, and the testing of many possible solutions.

Many of the problems of multiple users and concurrency described above can be handled with current technology. However, the provision of mappings has had very little previous research. This is the main focus of this research.

To provide the user interface we hope to utilise some of the tools developed at Auckland University in the Kea project [12, 13]. Kea is a language for the development of knowledge-based systems in the building industry, comprising a unique mix of functional and object-oriented languages. As part of the Kea environment there is an evolving set of tools which provide a forms interface and a CAD-like interface to the user (see Figure 3.). These tools are being improved to enable dynamic construction of forms and the CAD environment from a particular program. These Kea interface tools will provide the basic building blocks for the design views we require.

Figure 3. The Kea environment

In addition to being able to display differing views for differing purposes, we are also interested in maintaining consistency between the various views, so that modifications to any view are propagated appropriately to other views. This problem is directly analogous to work being performed in developing tools for visual programming environments. The MViews visual programming environment [7] provides an object-oriented framework for supporting multiple, overlapping, editable views of a program, with a built-in consistency management system which propagates modifications from view to view automatically via a common model of the program. Given the obvious parallels with the CBM work of ICAtect, we expect the lessons from the MViews work to have considerable impact on the next generation of the ICAtect system.

5. Conclusions

With the evolution of standard building models for the building industry it is necessary to provide user definable and modifiable interfaces.

Many of the technologies required to provide multiple definable user views have already been developed and there are many groups working towards this goal. However, little of the work has concentrated on a formalised schema for the definition of a user interface.

The development of the next generation of ICAtect will work on the definition of this methodology. It will also incorporate related work on visual programming, graphical user interfaces for code of practice conformance applications, and work on consistency management within a multiple-view environment.

Acknowledgment

The financial assistance provided by an Auckland University Graduate Scholarship is gratefully acknowledged.

References

[1] Luiten, B. and Tolman F. (1992). Computer Aided DfC (Design for Construction) in the Building and Construction Industries, CIB W78, Computer and Building Standards Workshop, Montreal, 11-15 May.

[2] STEP (1991). Part 1: Overview and Fundamental Principles, Draft N14, ISO TC 184/SC4/WG6.

[3] Bowen, J. and Bahler, D. (1992). Negotiation in Concurrent Engineering, Technical report TR-92-10, North Carolina State University, USA

[4] Wong, A., Sriram, D. and Logcher, R. (1992). SHARED: An Information Model for Cooperative Product Development, submitted to IEEE Computer.

[5] Clarke, J.A., Rutherford, J.H. and MacRandal, D. (1989). An intelligent front-end for computer-aided building design, University of Strathclyde, Scotland

[6] Batini, C., Lenzerini, M. and Navathe, S.B. (1986). A Comparative Analysis of Methodologies for Database Schema Integration, ACM Computing Surveys, 18(4), December, pp. 323-364.

[7] Grundy, J. and Hosking, J.G. (1992). The MViews framework for constructing multi-view editing environments, Proceedings New Zealand Computer Science Research Students' Conference, Hamilton, 28-30 October.

[8] Amor, R., Groves, L. and Donn, M. (1990). Integrating Design Tools: An Object-Oriented Approach, Building Systems Automation-Integration, First International Symposium, June 2-8, University of Wisconsin, Madison, Wisconsin, USA.

[9] Amor, R. (1990). ICAtect: Integrating Design Tools for Preliminary Architectural Design, MSc thesis, Department of Computer Science, Victoria University of Wellington, Wellington, New Zealand.

[10] Dearden, R. (1991). Object-oriented CAD system, Honours report, Department of Computer Science, Victoria University of Wellington, Wellington, New Zealand.

[11] Karstila, K., Björk, BC. and Hannus, M. (1991). A conceptual framework for design and construction information, Building Systems Automation-Integration, First International Symposium, June 2-8, University of Wisconsin, Madison, Wisconsin, USA.

[12] Hosking, J.G., Mugridge, W.B. and Hamer, J. (1991). An architecture for code of practice conformance systems, in Kahkonen and Björk (eds) Computers and Building Regulations, VTT Symposium 125, VTT Espoo, Finland, pp. 171-180.

[13] Hosking, J. G. (1990). A PlanEntry Package for Thermal Design, BRANZ contract No. 85-024, Technical Report No. 7, Department of Computer Science, Auckland University, Auckland, New Zealand.