SERIES BU1

AEN 31427
B23471

BRANZ

THE RESOURCE CENTRE FOR BUILDING EXCELLENCE

CI/SfB (A3):
(A5s)

REPRINT

No. 121 (1993)
Design Tool Integration: Model Flexibility for the Building Profession.

R. Amor, J. Hosking, R. Mugridge,
L. Groves, M. Donn

From: Proceedings of Building Systems Automation Integration 1992
Symposium on Computer Integration for the Building Industry,
Dallas, Texas, June 1992

ISSN - 0111-7459

Design Tool Integration: Model Flexibility for the Building Profession.

by

Robert Amor, Dr John Hosking
Auckland University
Lindsay Groves, Mike Donn
Victoria University of Wellington

Abstract

The development of ICAtect, as discussed in the Building Systems Automation and Integration Symposium of 1991, provides a way of integrating simulation tools through a common building model. However, ICAtect is only a small step towards the ultimate goal of total integration and automation of the building design process. In this paper we investigate the next steps on the path toward integration.

We examine how models structured to capture the physical attributes of the building, as required by simulation tools, can be used to converse with knowledge-based systems. We consider the types of mappings that occur in the often different views of a building held by these two classes of design tools. This leads us to examine the need for multiple views of a common building model.

We then extend our analysis from the views required by simulation and knowledge-based systems, to those required by different segments of the building profession (e.g. architects, engineers, developers, etc.) to converse with such an integrated system. This indicates a need to provide a flexible method of accessing data in the common building model to facilitate use by different building professionals with varying specialities and levels of expertise.

1. Introduction

ICAtect (Amor et al. 1990; Amor 1990) 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.

Though many of these design tools have been available to designers for years, there have been several obstacles to their acceptance and use. Amongst these are:

- 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.

- 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 error in describing the building to the tool.

- 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.

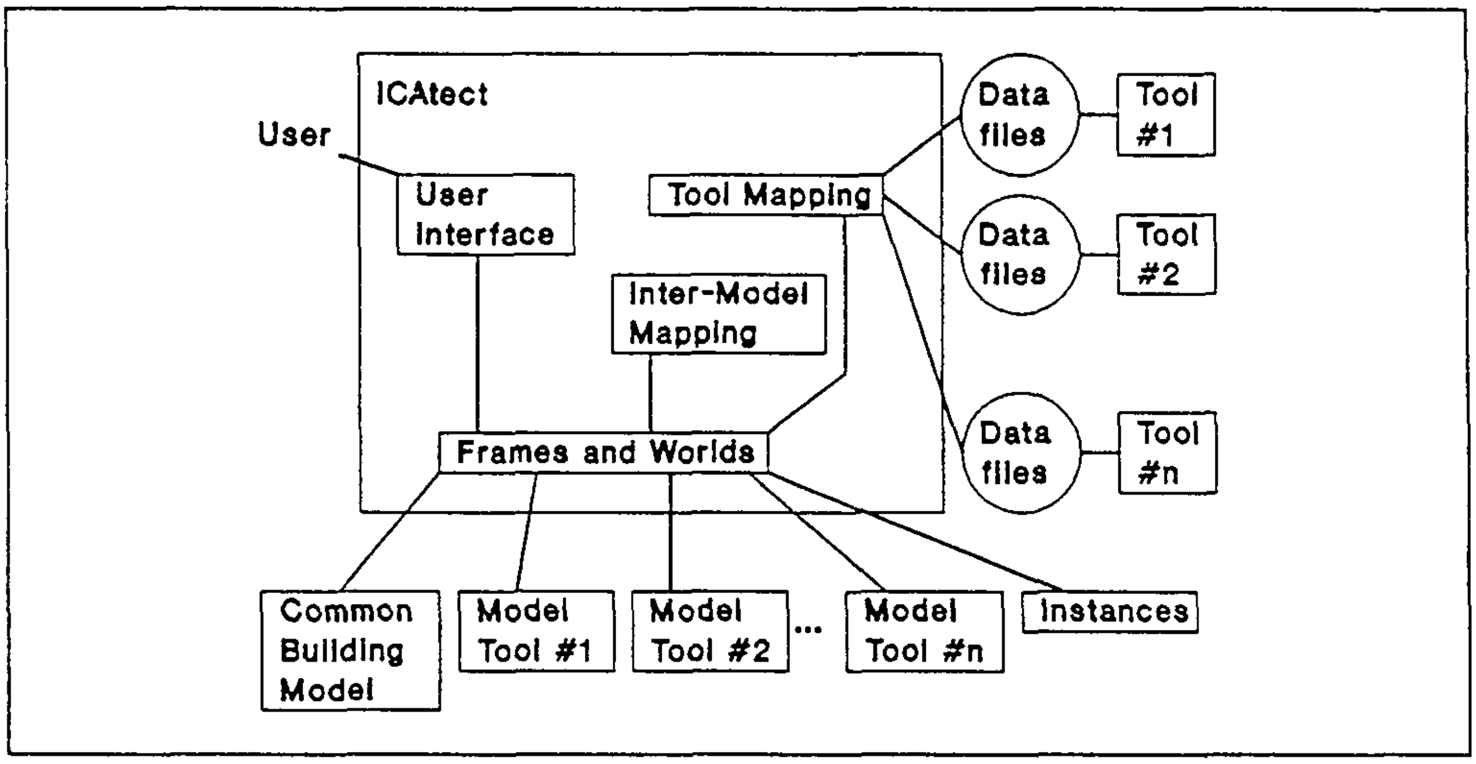

Figure 1 Existing structure of the ICAtect system

To date, we have addressed these problems with a prototype version of ICAtect (structured as in Figure 1) that allows the user to move information between a select number of design tools, and request information from the user as and when needed for simulation runs.

This paper discusses the further development of ICAtect, which is intended to provide greater useability to design professionals through the provision of multiple user views of the building model. The applicability of these views is being tested through the incorporation of knowledge-based systems into the existing ICAtect system. To give access to expertise and applications in the area of knowledge-based systems in building design, a collaborative partnership between the ICAtect group at Victoria

University of Wellington and the Kea group of the University of Auckland has been established. This partnership will advance the research aims of both groups as they explore compatible areas of automation and design.

In the following section we look at the development of the ICAtect system. Section 3 looks at the requirements for interaction with a building model, and Sections 4 and 5 consider the incorporation of new user and design tool classes into the CBM, with consideration of the requirements for interaction with a building model.

2. The Development of ICAtect

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.

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.

We also wish to allow the user to examine their 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 the prototype ICAtect system and work performed last year on a graphical interface to ICAtect (Dearden 1991) 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, i.e. 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 lack 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 (Karstila et al. 1991; Cherneff 1991; Clarke et al. 1989) demonstrates quite clearly that this project has much scope outside the field of preliminary architectural design for which it was first envisaged.

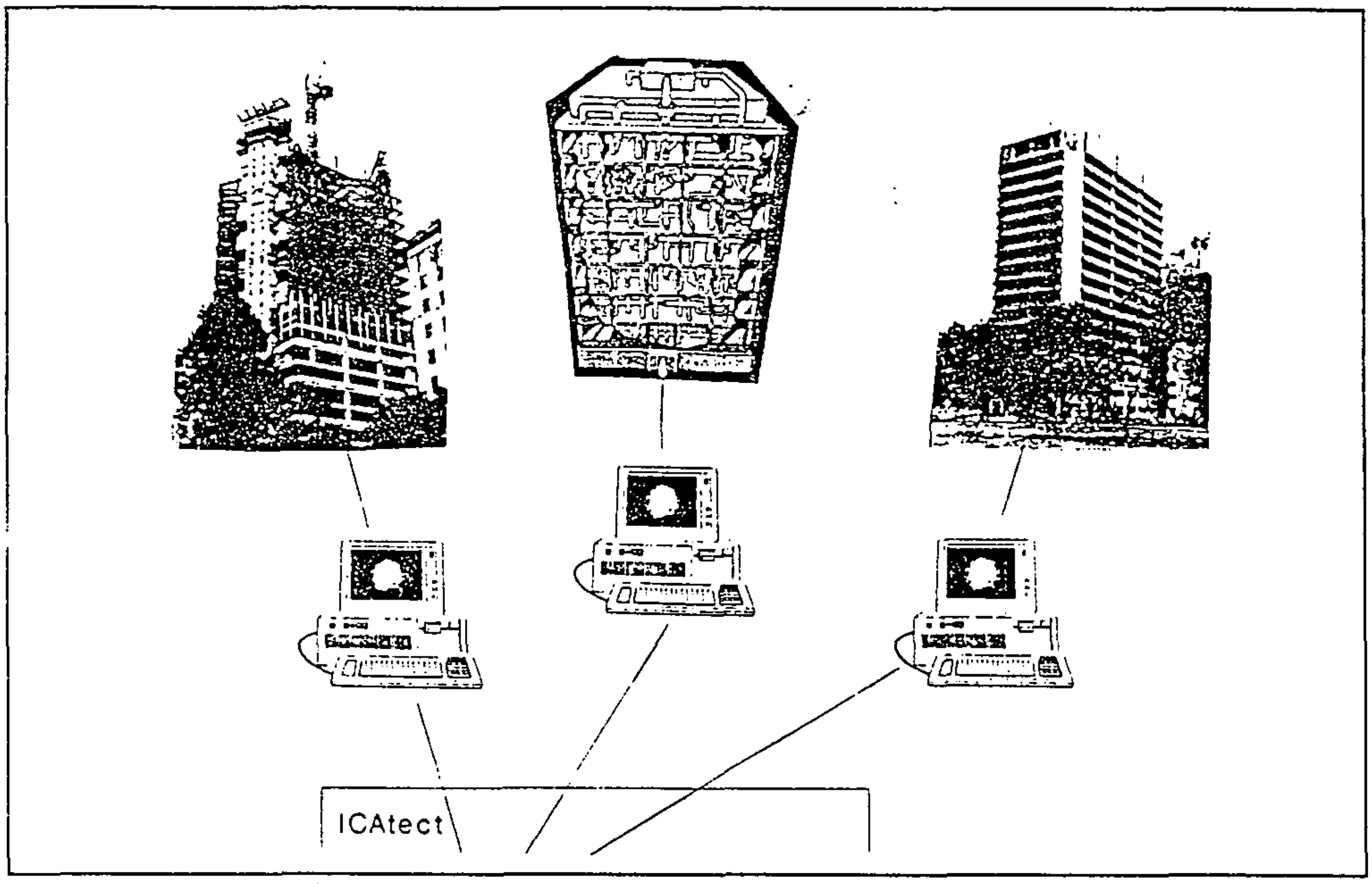

Figure 2 Multiple views into the ICAtect system

To address these problems we now generalise the structure of ICAtect's user interface subsystem to make it capable of handling multiple views of the building model. This structure (as depicted in Figure 2) allows 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. 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. This can be extended to opening up or shutting down particular sections of ICAtect to different classes of users, e.g., many architects may understand a general result of a structural analysis program but not its detailed output.

3. Requirements for interaction with a common building model

When originally devising the framework for representing building designs it was determined that the CBM would have to satisfy three major criteria. It should be:

- Comprehensive: The model should represent all the objects and information contained in a reasonable range of design tools.

- Modifiable: The model should be structured in such a way that new classes of objects and information required by new tools can be added without major modifications to its structure. Such additions should not require major changes to previously defined interfaces with design tools.

- Non-redundant: Data captured about an object should only exist in one place in the model, to minimize redundancy in the representation.

This ideology has given us a solid basis for providing multiple views to the model. In the construction of the CBM we have generated a model which holds all the information required for the existing views in the examined set of design tools.

The access required for interacting with the CBM is similar for the various design tools, and for the various classes of users. Fundamentally we are mapping one model of a building to another model of a building. In the case of ICAtect there is always one common model to map to, i.e. the CBM.

In considering the problems that can occur when creating a view to the CBM we look to the example of database schema integration (see Batini et al. 1986) which offers four ways of defining equivalence of attributes. To decide which definition to use we need to look at which kinds of correspondences occur within the views we are considering. The four definitions, with examples of their occurrences in this project are:

- Identical perspectives: Design tools or users have exactly the same viewpoints in modelling the same objects. This extends to the actual representation, the modelling constructs used and also to having the same perception of the object. For example, when representing a wall in DOE2 (1981) and ThermalDesigner (Amor 1991) the wall height and length are called 'Height' and 'Length' in both tools and describe the physical dimensions of the wall represented as floating point numbers.

- Different, but equivalent perspectives: Design tools or users have different viewpoints in modelling the same objects. However, the perceptions are still the same and are coherent. In Batini et al. (1986) this definition is broken into three distinct parts:

  - Behavioural: representations are equivalent if for one representation there is a corresponding alternative representation that has the same set of answers to any given query. For example when considering the thermal capacity of an object

DOE2 uses a 'U-value' whereas ThermalDesigner uses 'thickness', 'resistance' and 'capacitance'. Both give the same result, but the ThermalDesigner inputs can not be derived from the DOE2 'U-value'

- One-to-one mapping: representations are equivalent if instances can be put in a 1-1 correspondence. For example tools which require input in English units as opposed to those which require input in SI units.
- Transformational: representations are equivalent if they can be obtained by applying a set of atomic transformations. For example, representing wall positions as offsets from previous elements such as a space, versus absolute co-ordinates for the wall.

- Compatible specifications: Several combinations of attributes can model the same object equivalently. Though they will neither be identical nor equivalent, the attributes, perceptions, and integrity constraints are not contradictory. For example, the structural view versus the thermal view of a wall. The structural view could model the strength and elasticity of the wall while the thermal view models the thermal capacity of the materials.

- Incompatible specifications: Un-resolvable conflicts regarding names, types and integrity constraints which give rise to incompatibilities in their present form. There is a hope that this case will rarely be observed as practitioners have developed views at least partially compatible with other practitioners as they have collaborated on building projects over the years.

4. Design tool interaction

The design tools included into the CBM to date have been simulation tools for various disciplines. The emergence of numerous knowledge-based systems in the last few years signifies that they are prime candidates for inclusion into the system. In addition, through the research efforts of Hosking et al. (1990; 1991), the University of Auckland has developed considerable experience in the development of knowledge based tools for building design, particularly in the area of code of practice conformance. The two classes of tools are characterised as follows:

- Simulation tools model a building and its environment for simulation using the laws of physics. Therefore, simulation tools need to have a detailed description of the physical properties of the objects in the building that they are simulating. There are simulation tools in the area of thermal design, lighting, structural analysis and HVAC. An analysis of these tools lets us build up a very detailed picture of the physical properties that can be used to describe any object. Also, as simulation tools deal with forces acting on and between objects, we also get a complete picture of the types of connections possible between objects.

- Knowledge-based systems model an area of expertise on a particular problem area. The models of buildings used in knowledge-based systems are usually concerned with the appearance and behaviour of objects, while the models used in simulation tools are more concerned with precise physical description. The model is usually both more qualitative, replacing numerical measures with qualitative ones; and more approximate, replacing detailed calculations of attributes by typical values or by heuristic estimation. These estimations are often based on small numbers of other attribute values, and on replacing detailed model structure with abstractions, often qualitative. Both of these approaches are intended to "simplify" the model to permit conclusions to be reached rapidly and with the use of minimal amounts of user supplied data.

Analysing the views required by these two classes of tool in terms of the four definitions of equivalence of attributes considered in the previous section we see that they fall into two slightly different areas. As the simulation tools require a physically based model of a building it is usually the case that there is either a single method of representing a certain property of the building, or there are well known transformations between alternative representations. Thus most of the mappings fall within the definition of identical perspectives or equivalent perspectives.

Knowledge-based systems, on the other hand, tend to use abstractions, particularly qualitative and structural approximations, which are quite specific to the area of expertise of the system. Moreover, those abstractions are often not made explicit within the system (see for example, Hosking et al. 1987). Incorporating such systems into the CBM thus involves more compatible specification mappings and the occasional apparently incompatible specification mapping. The tasks required for mapping between different views include:

- Mapping qualitative attribute values to quantitative and vice versa. For example, mapping qualitative values "large", "average" and "small", for a specific attribute, such as floor-area, to typical values, ranges of values, or fuzzy ranges of values as appropriate.

- Making implicit attribute approximations explicit, and checking their validity. This involves identifying assumed values for attributes, and suitable margins of error for their acceptability against actual values. Such unstated assumptions are particularly important as they provide bounds on the validity of the conclusions of the knowledge-based system (Mugridge and Hosking 1988).

- Making implicit structural approximations explicit, and providing a mapping from the structural approximation to the underlying physical model. This involves uncovering abstractions used, particularly in the case of heuristic approximations. For example Hosking et al. (1987)

describe the analysis of a moisture diagnosis system to uncover the qualitative physical model implicit within it.

The first task is usually fairly straightforward. The second task is problematic from the point of view of completeness, i.e. determining whether all the implicit assumptions have been made explicit. The third task is the most difficult conceptually. In our experience, systems which appear to have incompatible specification problems often result from underlying and unstated structural approximations.

As a first step towards incorporating some of the above concerns into ICAtect, we are planning to extend ICAtect to interface to ThermalDesigner (Amor 1991), a knowledge-based system for checking conformance of a building design against thermal insulation code requirements.

5. User views and interaction

There seems to be general agreement that future CAD systems should work at a more generalised level, with objects such as spaces, to which the designer can relate to. To make an integrated system truly useful to a range of design professionals this must be extended to include modulating the conversation with the user, to match their mode of thinking about the various components that comprise the building being considered.

The different perceptions of concepts held by different groups in the building industry has been recognised as a problem for many years. Most people will know of the cartoon that appeared in the Architecture Journal depicting the views of a swing held by the different groups in the industry. This cartoon has also been used to describe the various groups in the computing industry (e.g. analyst, systems designer, programmer, user, etc.) and gives an indication of how universal the problem of communication between different user viewpoints is, and how the solutions from one area may be appropriate for another.

The problem of interacting with users in languages 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 and so do not address the needs of other users. The one notable exception to this has been the IFE project of Clarke et al. (1989) which provides explicit procedures to handle different classes of user at various levels of expertise through different form sets to represent the same concept. 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.

This is similar to the Forms manager developed at the University of Auckland for use with the Kea language (Hosking et al. 1991). The development of Kea was motivated by the need for tools for the creation of knowledge-based systems for the building

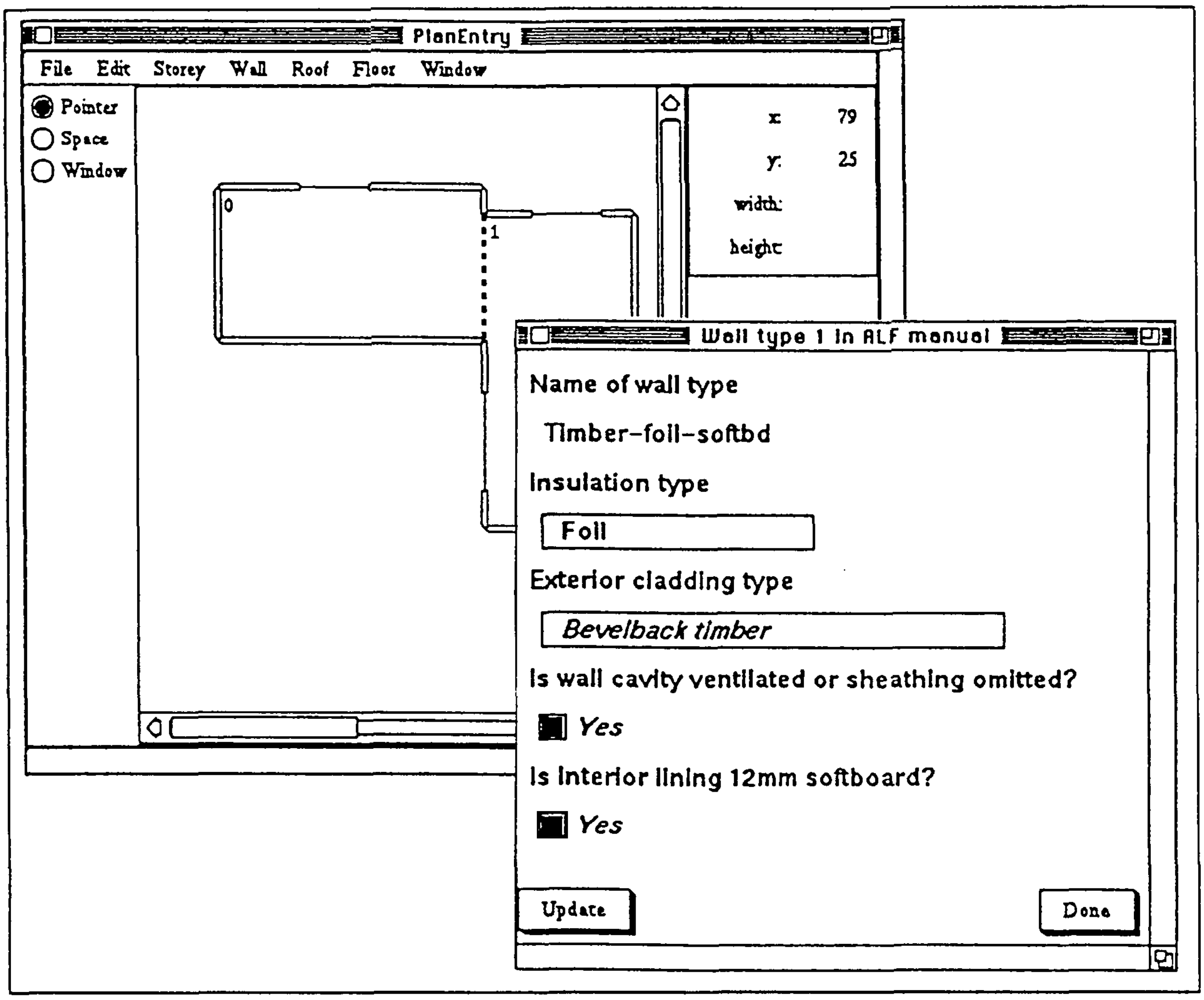

Figure 3 Forms manager and PlanEntry system

industry (Hosking et al. 1991). The Forms manager is used in conjunction with a graphical PlanEntry system, which enables the user to enter plan, or geometric information about a building design. This provides a simple set of tools capable of providing alternative views to different classes of user. Figure 3 shows a scene using the Forms manager and the PlanEntry system applied in ThermalDesigner.

Further development of the Kea graphics interface is in progress. The aim is to improve the flexibility and power of the graphics tools by integrating low level graphics capabilities directly into Kea. This will allow:

- The PlanEntry system to be constructed within Kea itself, rather than as a separate monolithic C-based package. The plan entry component will as a result be much more able to be tailored, with specialisations for different user views being readily constructed.

- Much more dynamic control over forms. Static form layouts will be constructed visually using a Kea-based "form painter", but these forms can be dynamically modified at run time to obtain views specialised to the needs of the user, thus allowing the system to respond directly to any perceived strengths or weaknesses of the user's view by modifying the form representing the concept being addressed. Figure 4 shows a mock-up of possible view variations.

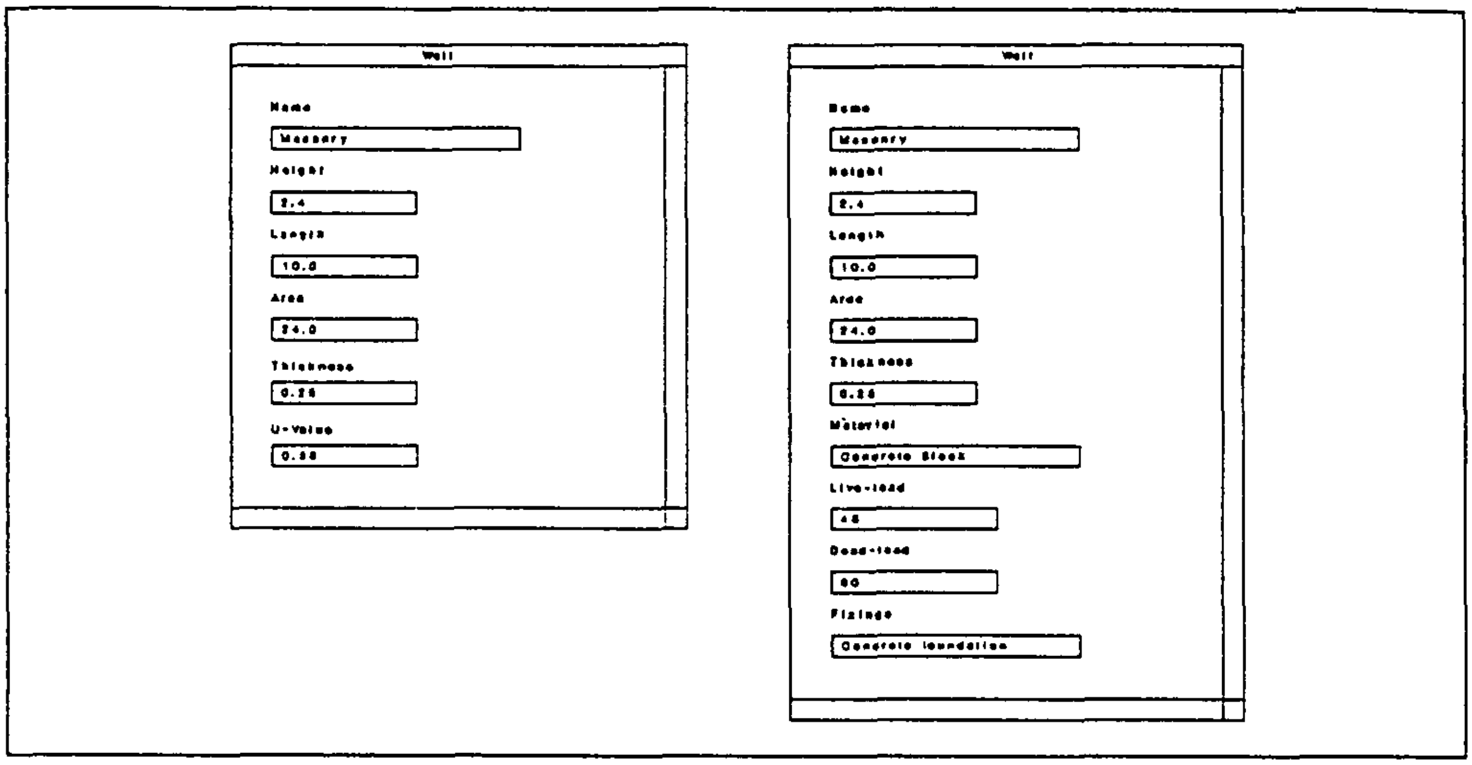

Figure 4 Thermal and Structural view variation mock-up in ICAtect

The Kea interface tools will thus provide basic building blocks for the design views we require. However, 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 by Grundy et al. (1991) and Grundy and Hosking (1992) in developing tools for visual programming environments. The MViews Visual programming environment 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.

6. Conclusions

The creation of ICAtect, integrating several design tools for preliminary architectural design, has highlighted the problems of providing a view of the resultant common building model to the various classes of design tools, and to different classes of users of the system.

To overcome these problems we start by looking at a more systematic way of classifying and representing the different views of the data that are required by the different classes of users and design tools. This classification system and integration methodology, drawn from work on database schema integration, is well suited to the problems encountered in the design tool and user view integration being attempted.

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

References

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.

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

Amor, R. (1991). ThermalDesigner, BRANZ contract No. 85-024, Technical Report No. 24, Department of Computer Science, Auckland University, Auckland, New Zealand.

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.

Cherneff, J. (1991). Integrating Design Data Schemata: the role of interpretation, Building Systems Automation-Integration, First International Symposium, June 2-8, University of Wisconsin, Madison, Wisconsin, USA.

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

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

DOE-2.1C (1981). Building Energy Simulation Group, Lawrence Berkeley Laboratories, University of California, Berkeley, California 94720, USA

Grundy, J., Hosking, J.G. and Hamer, J. (1991). A visual programming environment for object-oriented languages, in Korson, T., Vaishnavi, V., and Meyer, B. (eds) Technology of Object-Oriented Languages and Systems TOOLS 5 Proc of the 5th International Conference Santa Barbara, Prentice Hall, pp 129-138.

Grundy, J. and Hosking, J.G. (1992). Formal specification of the MViews visual programming environment, to be submitted to IEEE Trans Soft. Eng.

Hosking, J.G., Mugridge, W.B. and Hamer, J. (1987). A knowledge-level analysis of the Damp problem, Proc. 10th New Zealand Computer Conference, Christchurch, New Zealand, pp 29-40, 1987.

Hosking J.G., Hamer, J., Mugridge, W.B. and Dechapunya, A.H. (1990). From FireCode to ThermalDesign: KBS for the building industry, New Zealand Journal of Computing, 2(1), pp. 23-32.

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.

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.

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.

Mugridge, W.B. and Hosking, J.G. (1988). The development of an expert system for wall bracing design, Proc. NZES'88 The Third New Zealand Expert Systems Conference, Wellington, May , pp. 10-27.

BRANZ

THE RESOURCE CENTRE FOR BUILDING EXCELLENCE

BRANZ MISSION

To promote better building through
the application of acquired knowledge,
technology and expertise.

HEAD OFFICE AND RESEARCH CENTRE

Moonshine Road, Judgeford
Postal Address - Private Bag 50908, Porirua
Telephone - (04) 235-7600, FAX - (04) 235-6070

REGIONAL ADVISORY OFFICES

AUCKLAND
Telephone - (09) 524-7018
FAX - (09) 524-7069
290 Great South Road
PO Box 17-214
Greenlane

WELLINGTON
Telephone - (04) 235-7600
FAX - (04) 235-6070
Moonshine Road, Judgeford

CHRISTCHURCH
Telephone - (03) 663-435
FAX - (03) 668-552
GRE Building
79-83 Hereford Street
PO Box 496