Building Performance Simulation: Delivering the Power to the Profession

Clarke J A & Rutherford J
University of Strathclyde
Scotland

Mac Randal D
Rutherford Appleton Laboratory
England

Abstract

At the present time several powerful simulation models exist for the assessment of building environmental performance at the design stage. However when used as design tools these models suffer from several fundamental limitations. Typically they fail to tackle the problematic issues surrounding data preparation in the face of uncertainty. Invariably models are functionally orientated, containing little knowledge of the application domain. This means that they cannot direct a users' line of enquiry, allowing 'Why do you ask' type responses for example. Instead, the user must be expert enough to devise an appropriate performance assessment methodology and to coordinate model operation against this. These are the limitations being addressed in a project to develop an Intelligent Front End (IFE) for building performance simulation. The objective is the design of a machine environment which can act as an expert consultant to assist the user in the problem description phase, recognise the user's appraisal wishes, commission simulations and report back on overall performance.

This paper describes the form and content of the IFE which is conceived as an intricate synthesis of user modelling, human-computer interaction techniques, contextual knowledge and the interface to the various performance prediction models at its back-end. It also discusses the knowledge entities contained within, and manipulated by, the IFE.

Introduction

At the present time several building modelling systems are emerging which can address the range of cost and performance issues: from realistic visualisations, to a detailed appraisal of the operational performance of the energy, lighting and control systems. In response a number of different user types – from a design engineer or architect to a researcher or educationalist – are attempting to utilise these models, both at the early and later stages of the design process. The recent formation of professional bodies which represent the notion of building performance modelling (the International Building Performance Simulation Association [1] in North America and the Building Environmental Performance Analysis Club [2] in the UK), the advanced modelling programmes of the UK DEn [3] and US and UK research organisations [4, 5], the model validation and development activities of the European Community [6, 7] and the workstation and model acquisitions of the larger design organisations are signal events.

There remains, however, a major barrier to the effective, and routine, use of these modelling systems. The systems are invariably complex, requiring a high level of user expertise and, as yet, it has not been possible to incorporate a sufficient degree of intelligence within the models themselves with respect to problem description and performance assessment methodology.

To address these issues, the authors were awarded a grant by the UK SERC to develop a prototype intelligent front-end [8, 9], targeted, in the first instance, on building energy simulation. Such an IFE would act as a consultant to the user, conducting a dialogue to gather information on the design, identifying the user's objectives and commissioning the required performance simulations. In essence, the IFE would perform the mapping from the user's model of the energy program to the energy program's model of the user.

The IFE Architecture

The IFE is an intricate synthesis of user modelling, human-computer interaction techniques, contextual knowledge and the interface to the range of performance models at its back-end. It is built from cooperating modules organised around a communications module, the 'blackboard', to facilitate multiple use of information. Several other modules exist and can examine this blackboard for information, posting results back to it. Figure 1 shows the software modules comprising the IFE. These include:

- A *User Model* to track the user's progress and ensure the system responds in an appropriate manner.
- A *Dialogue Handler* to converse with the user in a manner which is tailored to her/his conceptual class and level of experience.
- A *Plan Recognition* module to identify the user's performance appraisal wishes.
- A *Planner* to determines the most appropriate appraisal methodology.
- A *Building Model* to create, from the information supplied by the user and the knowledge base, the data structure which describes the building under consideration in terms of its geometry, construction, occupancy and systems.
- A *Back-End Handler* to drive the required appraisal package, feeding it the necessary data. This module is the only one specifically tailored to a given program.
- And a *Knowledge Handler* to manipulate the knowledge concerning the application (building design), the domain (energy modelling for example), the appraisal software (ESP for example [10]) and the user class.

The functions to be handled by the IFE therefore include conversing with the user in the appropriate terminology, planning how to use the energy model to achieve the user's objectives, collecting and organising the description of the building, generating the necessary data and control input for the model and storing both the domain knowledge and the strategy knowledge to be used to drive the package.

The Blackboard

This lies at the heart of the system and has two major functions. Firstly, it holds the data representing the current state of the problem definition as input by a user or inferred by the knowledge handler. Secondly, it act as a communication centre for its various clients.

Data Storage

It is important to store data efficiently, so that the overhead of accessing is not great. This implies that the Blackboard should not only be capable of receiving and storing data but also of receiving arbitrarily structured information from a client and passing this on as a unit to other clients. However, given that clients are autonomous entities and therefore must not be expected to know the data structures created by other clients, it is also necessary for the blackboard to be able to search these structures internally and return that part of the data requested. Thus the data structuring must be **explicit**, that is generated by the client and stored with the data, **general**, that is able to handle any client data handling requirement, and **transparent**, that is other clients must be able to retrieve data without knowing the structure. At the present time the use of a schema definition language is being investigated.

Client Communication

Blackboard clients are divided into two classes: those at the user end, concerned with extracting from the user the design description and appraisal definition, and those at the back-end, concerned with creating a data structure suitable for use by the particular appraisal package. These two aspects are held within separate areas on the Blackboard and are stored as a series of attribute/value pairs. An attribute can be thought of as the name of a 'concept' which is meaningful to one or more clients. Usually, though not necessarily, these concepts correspond to concepts utilised by the user and/or appraisal package when describing this class of problem (for example, latitude, time-step or window width in an energy modelling context). There is no restriction imposed by the Blackboard on the contents of the value field of the attribute.

The Blackboard is not just a passive data structure. Clients can ask the Blackboard to create new, named areas. The various clients can then post items, that is attribute/value pairs, on any Blackboard area, the name of the poster and the time of posting being recorded with each item. Clients can also ask the Blackboard for the current value of any attribute by identifying the Blackboard area and the attribute name. Firstly, and most importantly, this avoids clients having to poll the Blackboard. Secondly, it provides a mechanism whereby two, or more clients can create a Blackboard area which acts as a communication channel between them. For example, this is used to pass information from the Dialogue Handler to its associated knowledge base (for user input validation for example).

Currently, clients are independent processes running asynchronously and the Blackboard does not impose any selection criteria on incoming messages. Should it become desirable or necessary to provide some form of scheduling/resource allocation for the various clients, an area on the Blackboard could be established to act as a control centre.

User Model

This module sets the appropriate user class (conceptualisation) and level within a class. This ensures that the IFE output is tailored to the user's skill level. It has the potential to monitor the dialogue with the user and the response of the user conceptualisation part of the IFE knowledge base. This allows it to ascertain the response speed of the user as well as the number of errors, system default overrides, changes of mind and backtracks. Note however that real, dynamic user modelling has not been attempted at the present time. Instead, the user is initially classified, from a database or explicit user input, and the corresponding user conceptualisation knowledge base initiated. (Currently there is only one user conceptualisation available, corresponding to a person who is relatively experienced in energy modelling, though several more are planned [11]). The user is then classified on an expert through novice scale so that appropriate level of guidance, feedback or help can be given by the main knowledge bases during the session.

Plan Recognition & Planner

At the present time, the user is explicitly asked to specify appraisal objectives in terms that the Planner module can handle. The Planner takes these objectives and decides what analysis to carry out by using a rule-based system which maps the user objectives onto one or more pre-defined, parameterised analysis scripts. The planner selects the values for the parameters based on the information available in the appraisal conceptualisation on the Blackboard. It also checks that the appraisal conceptualisation has sufficient information to generate the necessary data-sets and asks the Building Model module to construct them. The back-end is then asked to carry out these analysis.

Building Model

This module has the capability to establish an application specific data-set from the IFE's internal building model as held on the appropriate area of the Blackboard. This internal building model exists as an amorphous data-set which describes, in 3-D, the building geometry with attribution (thermo–physical properties) defined at the polygon level.

Back–End Handler

The Back–End Handler executes the script file as established by the planner and delivers the results back to the Blackboard for transmission to the user via the dialogue handler.

Dialogue Handler

This module handles all the interactions with the user. It passes messages and requests posted on the Blackboard to the appropriate applications software which comprises the actual user interface, posting back the user's responses. Thus the Dialogue Handler is essentially a communication switch and protocol converter.

The Dialogue Handler has several programs at its disposal to achieve interaction with the user. The primary mechanism is via a generic Forms Package which can manipulate a set of forms which correspond to a given user conceptualisation. Each field on a form is considered to be a 'concept' which corresponds to a concept in the user's mental model of the appraisal domain. Via the Forms Package, the user can ask about these concepts as well as associate values with them. The Forms Package is described in more detail below. The other facilities for interaction include a map display for positional inputs and a geometric modelling package for geometry definition and perspective feedback.

Knowledge Handlers

There are a number of separate Knowledge Handlers in the IFE, corresponding to the various domains being addressed. Basically, each one is an autonomous inference engine associated with its own knowledge base. Currently, all the inference engines are Prolog interpreters, though it would be simple to interface to other inference engines, for example an expert system shell. In essence a Knowledge Handler is an interface to the Blackboard which converts the syntax of the posted items into a form suitable for internal manipulation and vice versa. For example, for a Prolog interpreter, the items passed to it by the Blackboard have to be converted into Prolog goals (predicates) and the prolog syntax has to be filtered out before passing back new items to be posted.

There are three knowledge bases (KB), dealing respectively with the two problem definitions as held on the Blackboard and the user dialogue itself.

User Conceptualisation KB

This KB builds, on the Blackboard, the problem definition as perceived by the user (the user conceptualisation). Given what is already known about the problem and the user's objectives, it can deduce what 'concept' values are sensible (that is intelligent input validation), and how to derive appropriate default values for concepts, (that is intelligent defaults).

Appraisal Conceptualisation KB

This KB builds, on the Blackboard, a problem definition suitable for the back–end appraisal package, (the appraisal conceptualisation). It knows what data (appraisal 'concepts') are required to carry out the requested analysis, requesting missing data from the user conceptualisation or intelligently defaulting it as appropriate. It also knows where and how to find any supplementary data by accessing standard databases.

Dialogue KB

Working in close conjunction with the Dialogue Handler is a Dialogue KB. The communication between the two is via a dedicated Blackboard area (which can be monitored by the user model). This KB handles the dynamics of the user interaction. Its main function is to provide feedback, help and guidance to the user, and to this end it contains knowledge about the capabilities of the Forms Package, the other interaction facilities and the formset and fields that are being used in a given session. It also monitors user conceptualisation area of the Blackboard for the context sensitive default values suggested by the user conceptualisation KB and passes them on to the Forms Package.

The Forms Package

The Dialogue Handler is responsible for managing and monitoring user interaction; deciding what level of communication is necessary or appropriate for a given user or task. Unlike conventional systems where the interface remains static throughout a session, by keeping track of a user's progress, the Dialogue Handler can, in conjunction with the User Model and Knowledge Handler, tailor the dialogue to the user's level of expertise and performance history. The Dialogue Handler utilises a general purpose Forms Package to coordinate the two-way user interactions.

The objective of the Forms Package is to manipulate a set of forms, passed to it by the Dialogue Handler at the dictates of the User Model, which correspond to a given user conceptualisation. Any single form is a collection of logically related concepts, each concept being represented graphically by an identifier and an associated answer field. It is important to appreciate that the Forms Package does not impart meaning to any user entries. It merely passes on all entries to the Knowledge Handler.

Each form (or form-set) is defined by a template file which establishes the form concepts and their relative positions on the form. This technique allows any user conceptualisation to be established as a given form-set for use with the IFE. All that is then required is that the related knowledge entities be established and entered in the user conceptualisation knowledge base.

Forms can be nested to categorise concepts and to provide scrolling regions. Figure 2 shows an example form and its associated template file.

All user interaction with a form is controlled by mouse and keyboard inputs. The former is used to select a concept, the latter to make an entry. Two pop-up menus are associated with each field, each activated via the mouse. The first is a general menu offering a number of general facilities including requests for help and example entries, or requests to the IFE to provide an intelligent default. The second is a concept specific menu listing the options on offer at any given point in time.

The Forms Package utilises dynamic memory allocation and is designed to operate on the basis of instructions received from the Dialogue Handler. This means that the status of any field (or form) can be changed at any time. For example, the options offered on a menu or the state of a particular field – displayed or hidden – can be dynamically set by the IFE. In particular, this allows the user conceptualisation to be shifted, up or down, at the dictates of the User Model. It also facilitates several advanced features, including animated graphics and sub-form 'tear-off' and exporting.

IFE Future Work

At the current time, a prototype IFE is operational within a Unix workstation environment and several refinements are planned or underway.

Several additional user conceptualisations are being researched, with the focus on the more design oriented, and less technically motivated, users such as Architects. The corresponding form-sets will then be developed and the associated knowledge entities installed in the user conceptualisation knowledge base. The User Model will also be extended so that different levels of users in each class can be handled. The plan recognition module is being enhanced to address the wider, less focused objectives typical of such users.

The building model is being generalised so that it can cater for a diverse range of performance appraisals: from energy and comfort considerations, through acoustic and lighting performance assessment, to the production of fully animated, three dimensional images.

The associated knowledge entities to handle this extra information, in terms of intelligent defaulting, validation and the various subsets of the IFE's internal building model that are required by different user objectives and appraisal methodologies, is being added to the knowledge bases.

And, finally, the range of rule-based, performance assessment methodology on offer by the IFE is being extended.

Conclusions

It has been found that, for all but the most knowledgeable and experienced designers, assistance with current CABD software is essential. One of the advantages of an IFE in the Architectural and Engineering field is that, for typical designers, help in dealing with the varying scientific, engineering and design vocabularies would be very valuable. It would substantially increase the rate of uptake of sophisticated appraisal packages.

The main outcome of the IFE project will be a generalised IFE which can be interfaced to any existing appraisal package. This will open up the possibility of truly integrated, multi–criteria design appraisal. This, in turn, will allow the designer to more easily handle the various trade–offs necessary in the design and re–design of buildings, helping to improve the quality of the built environment by allowing the designer to eliminate poor performance features at the design stage.

References

1. 'IBPSA Bylaws and Charter Statement' Available from: Sowell, E., Department of Computer Science, Cal State Fullerton, Fullerton, California.
2. 'BEPAC Rules of Association' Available from: Keeble, E., Building Research Establishment, Garston, England.
3. Energy Technology Support Unit 'Project Brief: Performance Analysis Service' Harwell, Didcot, 1987.
4. Winkelmann, F 'Developments in Object–Orientated Programming in the Area of Building Performance Appraisal' Various Publications, Simulation Research Group, Lawrence Berkeley Laboratory, Berkeley, California.
5. Clarke, J A et al, D 'Proposal to the UK SERC to Develop an Energy Kernel System for Building Energy Simulation' Energy Simulation Research Unit, University of Strathclyde, Glasgow, 1988.
6. Gicquel, R 'Programme PASSYS: Status Report' Ecole des Mines, Sophia Antipolis, 1987.
7. Hatten, D van (ed) 'Proceedings of a Workshop on the Future of Building Performance Modelling' EUR Report Reprint, March 1988.
8. Mac Randal, D 'The Application of Intelligent Front Ends in Building Design' pp. 360–370 in *Artificial Intelligence in Engineering, Tools and Techniques* eds. Sririam, D. & Adey, R. A., Computational Mechanics Press, 1987.
9. Mac Randal, D and Clarke J A 'A Proposal to Develop an IFE for Building Energy Simulation' Science and Engineering Research Council, Grant Case for Support, 1986.
10. Clarke, J A *Energy Simulation in Building Design* Adam Hilger Ltd, Bristol and Boston, 1985.
11. Clarke, J A et al 'Project IBIPS: Intelligent Building Input for Performance Simulation' A Research Proposal to the Commission of the European Communities under the Joule Programme of DGXII, 1989.

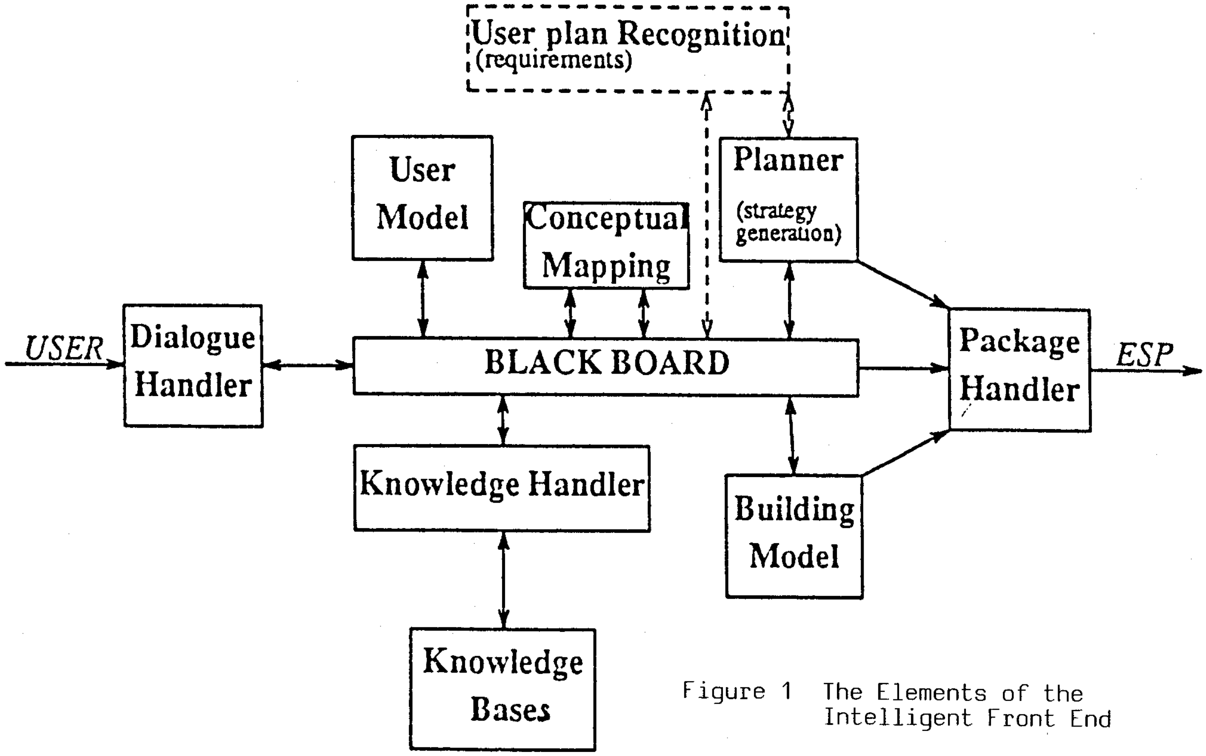

Figure 1 The Elements of the Intelligent Front End

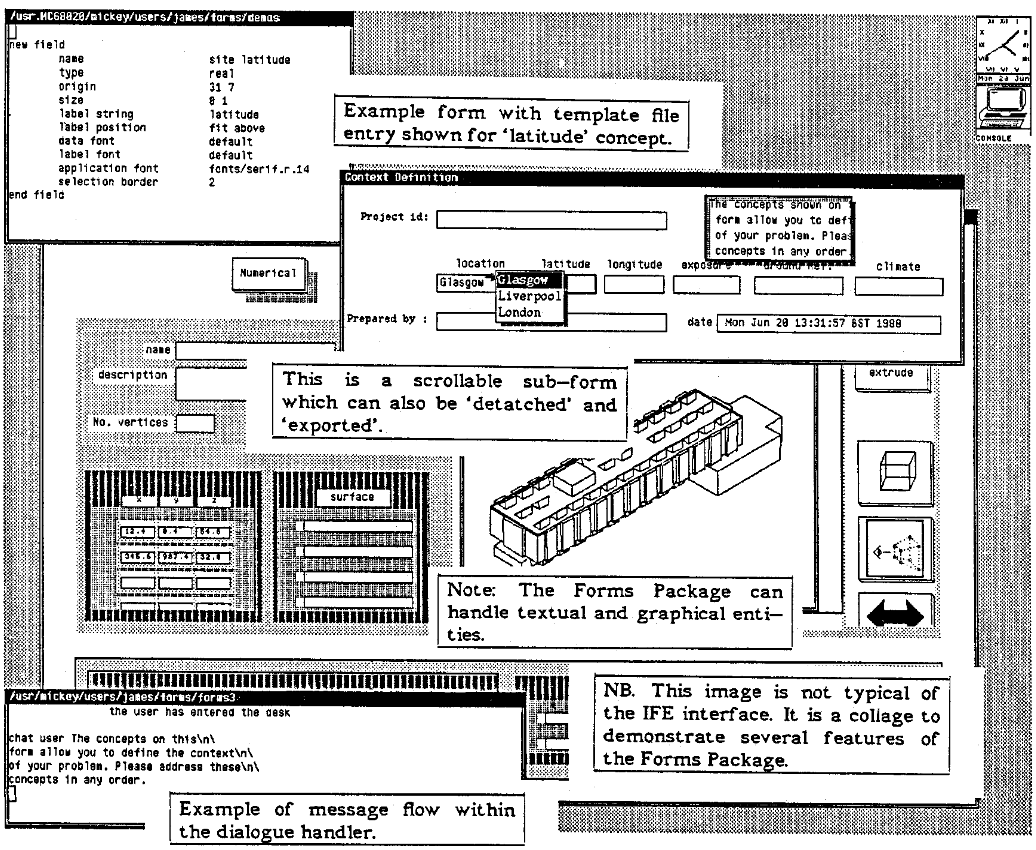

Figure 2 Capabilities of the Forms Package