Intelligent CAD as an Interface to Thermal Simulation Programs

Robin Drogemuller *
Harry Suehrcke **
James Cook University of North Queensland

Abstract

Thermal simulation programs for buildings have been available for many years. They vary from simple methods that can be performed at the sketch design stage using a spreadsheet through to sophisticated dynamic thermal simulations requiring mainframe computers. However, these methods are not widely used by building designers during the early stages of design when the most important decisions are made. Explanations for this are that the preparation of the data can be time consuming and few building designers are confident in interpreting the results of the simulations. One method to encourage designers to use thermal simulation tools 4 to incorporate these tools in the drawing environment that they use. Simulation programs will then be seen as part of the design process rather than an appendix.

This paper describes a research project, TES (Thermal Expert System), that integrates the graphical methods with which building designers are comfortable and thermal simulation tools. Work is currently restricted to detached houses. The user interface is based on an intelligent CAD system which only requires the designer to enter the features of the building which differentiate it from an archetype. The major task in developing this system was the implementation of the intelligent database (knowledge base) representing the building. This knowledge base had to store information informs that were suitable for interactive manipulation by the user and provides the information required for the thermal simulations. The integration of an intelligent CAD interface with thermal simulation programs allows thermal analyses to be more easily carried out while designing a building and provides either detailed or non-technical information as requested by the user.

Introduction

The energy usage in buildings is a significant component of the total energy usage in developed countries. Domestic energy consumption for heating and cooling in Australia accounts for approximately 10% of the total primary energy use(Ballinger *et al*, 1992). The total amount of energy used for domestic heating and cooling would be higher in countries with less equitable climates. Consequently, improvements in the thermal efficiency of housing design would have a direct impact on total energy demand.

The thermal simulation of buildings is normally performed by calculating the flow of heat through the building elements. If constant interior and exterior temperatures are assumed the analysis is steady state. If changes in environmental parameters and the effect of thermal mass are considered, the analysis is dynamic. The analyses can be used at two different levels. The building designer needs guidelines to assist in the choice of structure and building fabric, while the mechanical engineer needs accurate analyses to calculate the expected range of heating and cooling loads that will need to be serviced by the mechanical plant.

Building designers are not normally confident in using thermal simulation tools. This problem is compounded in Australia, since professionally trained designers are responsible for less than 5% of the housing stock constructed each year. The development of new methods for thermal analysis is not likely to improve their usage. However, the increased usage of CAD systems in building design and documentation provides an opportunity to combine the design and analysis of buildings. The graphical computer representation of the building can be used to provide the data required for thermal simulations if appropriate programs are developed. This does not solve the problem of the interpretation of results. Knowledge based methods provide a means of providing non-numerical information back to the designer after the simulation produces results.

---

* Department of Civil & Systems Engineering
** Department of Mechanical Engineering
James Cook University
Townsville Q 4811
Australia
Tel : (077) 814476
Fax : (077) 751184
Email : csrd@manta.icu.edu.au

Building Design and Thermal Simulation

Since the oil crisis in the 1970's significant research has been carried out to provide methods for building designers to examine the energy implications of their design decisions. Numerous charts and graphical design aids are available as are a range of computer based thermal simulation systems. However, even the simplest methods have not achieved a high level of penetration into the offices of building designers. The simulation programs are normally used after the design is substantially complete to fine-tune the proposals and to provide the data necessary to design the heating and cooling plant. This is normally too late in the design process. Ferry and Brandon (1991) claim that 90% of the decisions affecting the cost of buildings are made even before the first sketch designs are produced. Since these early decisions define the configuration, shape, structure and fabric of the building, the same can be said of the thermal performance of the proposed building. A method of comparing the thermal performance of competing building proposals is required to ensure that the initial decisions are made with the maximum available information. Since energy usage is a major component of the recurring expenditure for a building this should be considered an essential part of the feasibility study of a building.

Why are designers not using the tools that are available? One possibility is that the use of analysis tools is seen as too much trouble, particularly when the client is not concerned about thermal performance or deadlines are tight.

How can thermal simulation programs be incorporated into the design process so that they are not seen as a luxury? The increased use of CAD systems in architectural and drafting offices is a possible solution. If thermal simulation programs can be integrated with the standard CAD drawing tools then the simulations can become part of the process, rather than a separate constraint added to the many competing for the designer's attention.

As a building is designed the amount of information available about it increases gradually. When the feasibility of the project is being assessed very little specific information is available. As stated above, it is at this stage where most of the important decisions are made that affect the thermal performance of the building. Initial targets are normally set by examining the performance of similar buildings under similar climatic conditions. This allows preliminary sizing of equipment and plant rooms so that the design can start. As the design progresses more information about the particular project is available and more detailed assessments of the building's likely performance can be made. Default values can be provided at the start of the design process to allow simulations to be performed. As available information on the project becomes more specific further simulations can be run to refine the initial assumptions.

Clarke and Maver (1991) have previously suggested that intelligent front ends to thermal simulations would be worthwhile. This paper describes the use of a simple CAD system as an intelligent front end to thermal simulation programs. The original work lies in linking the intelligent front end to the thermal simulations. Currently the actual simulation programs used are of peripheral interest. A similar program called Energy Scheming (Brown & Sekiguchi, 1989) acts as a 'black box'. The designer is restricted to the one simulation method which is integral with the program. Energy Scheming does not allow for various levels of detail in the thermal simulations.

The following criteria were defined to guide the development of TES (Thermal Expert System) :

1) Run on commonly available computers.

TES is being developed to run under Windows on IBM PC's and on Macintosh computers.

2) Simple User Interface

The provision of interactive graphics to allow the entry of line drawings means that it is no longer necessary to produce the drawings using a CAD system and then extract the information required for thermal simulations. The required information is passed across to the thermal simulation programs as required.

3) Readily available estimates of energy usage

Estimates should be available throughout the design process, from outline proposals through to detailed designs. Default reasoning techniques are used to provide 'best guess' values for the thermal simulations as the building proposal is refined.

4) Minimise Data Entry

Textual information is entered through dialogue boxes with the available selections shown where this is appropriate. The user only needs to select the desired response. This minimises typing and errors.

5) Modularity

Different simulation programs are required at various stages through the design process and are required by different people. The independent development of different simulation programs for various purposes means that users only need to install the software appropriate to their needs. Modularity also simplifies the maintenance and extension of TES.

Using TES

When TES is started (figure 1) the user is presented with a screen consisting of three areas. The graphics window is the largest and contains drawing tools on the left hand side. The window at the bottom of the screen is provided for textual messages which should not interrupt the flow of work. A menu bar appears across the top of the screen to allow the user to issue commands.

When 'New' is selected from the 'File' menu a series of dialogue boxes appear asking questions about the building. Information about the client, the site, the building and the rooms within the building is requested in the order given. Information about the construction and finishes is then requested (Figure 2). Dialogue boxes are used rather than the text window to enter all of this information since the range of values that TES 'knows' can be presented with default choices already highlighted. Once the initial building information has been entered TES can generate advice on the suitability of the initial choices based on guidelines used for the various climatic zones. It is also possible to perform a steady state simulation by using typical default values for building size based on the area and likely plan shape of the house.

Once the parameters for the project are entered the user can then draw the floor plan of the building in the graphics window. The plan is built up by selecting a room name from the 'Rooms' menu and a graphic tool from the side of the floor plan window (figure 1). The rectangle or polygon drawn on the screen is accepted as the room and the lines around the perimeter are recognised as walls. TES automatically distinguishes between internal and external walls and stores the appropriate construction type for each element. When the outline plan is complete, TES can then generate the roof plan using the requested roof overhangs following one of the standard roof shapes. This also allows TES to extend the walls up to the underside of the ceiling. The doors and windows can then be added to the floor plan by the user. The elevations are generated automatically using default values for the window and door head heights and by calculating the height required for walls.

When the building geometry has been defined the thermal performance of the house can be analysed using either steady-state (Szokolay, 1990) or dynamic (Harrington-Lynn, 1974) simulations.

Implementation

The basic design concept used in knowledge based systems provided the philosophy behind TES (Maher, 1987). The knowledge about buildings, simulation methods and interpretation of results are all separated from the control knowledge. The knowledge based shell performs a coordinating role, requesting information from the various components of TES, passing requests for information between other components and displaying the results to the user. The intention of this separation of control from knowledge is that the knowledge sources can be updated and changed without having to modify the engine driving the modules.

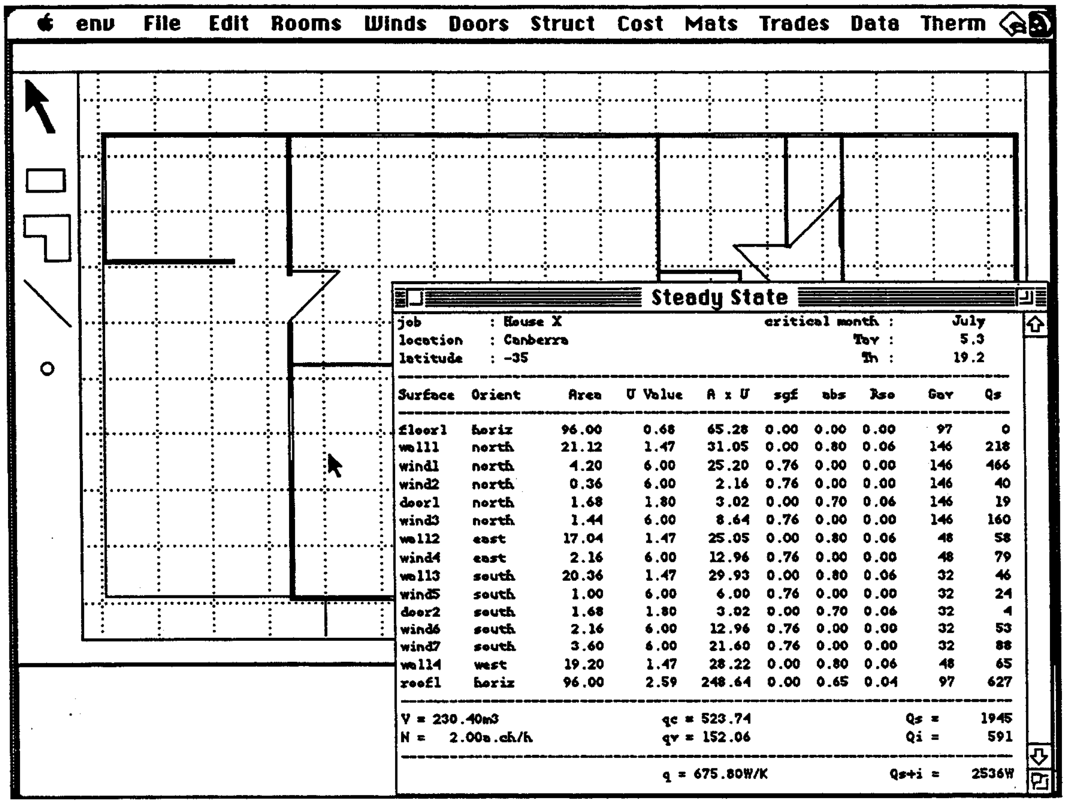
TES is written in PROLOG (Clockson & Mellish, 1984), a language used extensively in artificial intelligence programming since the methods of implementation rely heavily on artificial intelligence techniques. Flex (LPA, 1989), a knowledge representation language which is implemented in PROLOG is used for large parts of the programs. A range of expert system tools are available which can simplify the process of developing knowledge based systems but these were passed over since none of the options examined provided interactive graphics facilities suitable for the graphical components of TES.

A modular structure is used to implement the components of TES (Figure 3). Each module is a stand-alone program that is called by and communicates with the TES shell and the other modules. The shell is responsible for all interaction with the user. When it needs data from one of the other modules the shell sends a message to the appropriate module, which performs whatever calculations are necessary, exchanges messages with the other modules and returns an answer. Communication occurs through Inter Application Communication (IAC) supported by the operating system. The modular structure was chosen to allow incremental development of the system and to allow for alternative sources of data. The use of a knowledge base in such a manner is known as a blackboard system (Englemore & Morgan, 1988).

A typical sequence of interaction between the modules is for 'Shell' to send a command to 'TES' asking for a steady state simulation. 'TES' then passes this request to 'QBalance' which then requests a list of all of the external building elements and required properties for each element from 'Shell'. The construction type, area, orientation and shading for each area are sent. The climatic data for the location and the thermal properties of the elements are then obtained from the thermal database and the simulation is run. The results are returned to 'TES' which uses rules to interpret the numerical data and to derive recommendations. Any data and recommendations requested by the user are then returned to 'Shell' for display.

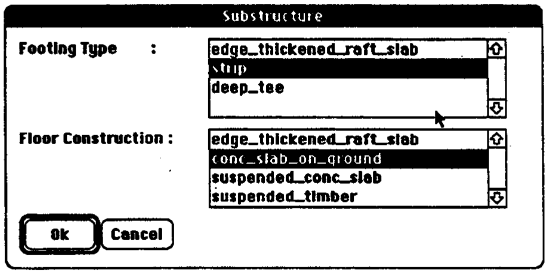
Knowledge Representation

TES is based on storing both knowledge and data. Computer programs manipulate data - numbers, words or graphics. TES manipulates all three. TES also manipulates knowledge such as the location of walls and the windows located in them. Data structures called frames (Minsky, 1985) are used to represent the knowledge about buildings within TES. An example is :



```
job          : House X                       critical month :      July
location     : Canberra                                 Tav :       5.3
latitude     : -35                                       Th :      19.2
---------------------------------------------------------------------------
Surface  Orient     Area  U Value   A x U   sgf   abs   Rso    Gav    Qs
---------------------------------------------------------------------------
floor1   horiz     96.00    0.68    65.28  0.00  0.00  0.00     97     0
wall1    north     21.12    1.47    31.05  0.00  0.80  0.06    146   218
wind1    north      4.20    6.00    25.20  0.76  0.00  0.00    146   466
wind2    north      0.36    6.00     2.16  0.76  0.00  0.00    146    40
door1    north      1.68    1.80     3.02  0.00  0.70  0.06    146    19
wind3    north      1.44    6.00     8.64  0.76  0.00  0.00    146   160
wall2    east      17.04    1.47    25.05  0.00  0.80  0.06     48    58
wind4    east       2.16    6.00    12.96  0.76  0.00  0.00     48    79
wall3    south     20.36    1.47    29.93  0.00  0.80  0.06     32    46
wind5    south      1.00    6.00     6.00  0.76  0.00  0.00     32    24
door2    south      1.68    1.80     3.02  0.00  0.70  0.06     32     4
wind6    south      2.16    6.00    12.96  0.76  0.00  0.00     32    53
wind7    south      3.60    6.00    21.60  0.76  0.00  0.00     32    88
wall4    west      19.20    1.47    28.22  0.00  0.80  0.06     48    65
roof1    horiz     96.00    2.59   248.64  0.00  0.65  0.04     97   627
---------------------------------------------------------------------------
V = 230.40m3                  qc = 523.74                 Qs =     1945
N =   2.00a.ch/h              qv = 152.06                 Qi =      591
---------------------------------------------------------------------------
                              q = 675.80W/K             Qs+i =    2536W
```

Figure 1 User Interface with Simulation Results



Footing Type : edge_thickened_raft_slab / strip / deep_tee (selected: strip)

Floor Construction : edge_thickened_raft_slab / conc_slab_on_ground / suspended_conc_slab / suspended_timber (selected: conc_slab_on_ground)

Figure 2 Typical Dialogue

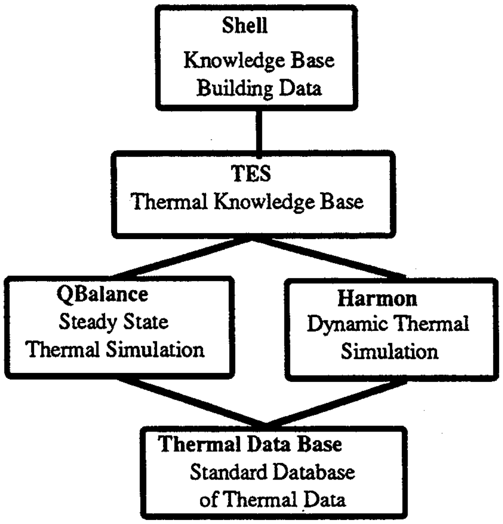


Figure 3 Structure of TES

```
frame window ;
   default frame_type is aluminium and
   default opening_type is
horizontal_sliding and
   default glazing is single_6mm and
   default head_height is 2100 and
   default sill_height is 900 and
   default width is 1500 and
   default height is 1200 and
   default u_value is 6.0 .
```

The various frames representing particular sizes of window are linked to their ancestor, the window frame. Any values for slots which are not explicitly over-ridden are inherited from the window frame :

```
frame window_1200x1200 is a window ;
   default width is 1200 .
```

The width slot (field) for this frame is now 1200 mm. All of the other information given in the window frame is inherited by the window_1200x1200 frame. This is indicated by the 'is a window' link.

Frames as implemented in Flex are a more powerful method of representing information than the standard methods in traditional programming languages. The most important characteristic of frames is the inheritance of information from parent to child frames as explained above. Frames can also have procedures attached to them. Launch procedures are used to call dialogue boxes which allow the user to set the appropriate values for the slots when a new instance of a frame is created. The value that is inserted into a slot can be controlled using constraint procedures to ensure that only legal values are stored. Demon procedures are attached to slots and are called immediately after a slot value is updated, usually to ensure that dependant information is also updated. When a new instance of a window is created the launch procedure is run and the new values are inserted into the slots of the instance. The values of these slots can be checked by constraint procedures as the values are inserted. Access to slots can also be controlled using watchdog procedures. Watchdogs are used to verify that the existing value in a slot is legitimate.

```
launch window
   when Window is a new window
   then ask window_height
   and ask window_width .

question window_height
   Please enter the height of the window ;
   input H such that integer(H) and H > 0.

question window_width
   Please enter the width of the window ;
   input W such that integer(W) and W > 0.

watchdog window_sill_height
   when the sill_height of Window is
requested
   then check that
valid_sill_height(Window) .

action valid_sill_height(W)
   do W`s sill_height := W`s head_height -
W`s height .
```

Frames are used to represent the building elements. For example, the external walls are stored as wall panels which intersect at changes in direction, junctions with internal wall panels and changes of material. If the wall panel is of the standard external wall material then no explicit reference is required. If a panel is of a non-typical material then a slot is created giving this information. When the material type is required for simulation purposes, the frame is queried. If there is an explicit type of material stored, then it is returned. Otherwise, the standard value is retrieved from the building frame and returned.

Frames are suitable for storing and maintaining descriptive data, but they are less useful for storing procedural information. Flex provides this facility with its rules :

```
rule window_east_orientation
   if the orientation of a window is east
   and the window`s area is greater than 1
   then write('The area of the window is
too large for an east facing window')
   and nl .
```

The extra facilities offered by frames and rules are not necessary in all circumstances. Where information in Shell or TES is likely to be modified by non-programmers, flex has been used due to its English-like syntax. Internal data such as the thermal properties of materials is stored as Prolog predicates to aid efficiency of execution. In their simplest form these are represented as :

```
u_value(window,metal_frame_6mm_single,
                                    6.00).
```

Data statements in Prolog give the type of data outside the brackets with the building element type, the construction type and the actual U value stored within the brackets. This format is equivalent to the standard relational data base. However, this is not adequate to represent all of the information required. The U value of some building elements, such as floors, depends on the geometric properties of the element. The calculation of the U value once the geometry of the element is known is done by procedures which define relationships between the items of data :

```
u_value(floor, Con_Type, U) :-
                     % with the floor element
   element(floor,_,_,L,W, Con_Type),
                     % find length & width
   flr_u_val(Con_Type, Lth, Wth, U_Val),
                            % get the U value
   L > Lth,
   W > Wth, !,                  %
```

```
        % check for appropriate values
  U is U_Val .              % got it

flr_u_val(suspended_timber, 30, 15, 0.39).
flr_u_val(suspended_timber, 15, 15, 0.45).
flr_u_val(suspended_timber, 15,7.5, 0.61).
```

Procedures in Prolog use the same form as data statements, but in this case the Con_Type and U are variables, recognised by their upper case first letter. The variables are assigned values by searching the Prolog database for statements which match. In this case, the length and width of the floor are looked up, and the first data item is read. If the length and width for the data item are acceptable the U value is returned. If the length and width for the current U value are not acceptable, the next is looked up, until an acceptable match is found and returned.

The heuristic information ('rules of thumb') used has been derived from a number of text books related to thermal design and simulation (Markus & Morris, 1980), (Ballinger *et al* , 1992). The computer based simulations were adapted from existing programs written in Pascal (Szokolay, 1992)

Geometrical Data

The Prolog procedures which store geometrical information must allow the data to be represented in a recognisable form in the graphics windows and must also allow for the extraction of areas and lengths for further processing. In order to maintain acceptable performance the following constraints applied :

Minimise computational load by saving intermediate results where possible;

Distribute processing between pauses by the user to maintain response times;

Keep representation simple but adequate; and

Minimise entry of lines.

TES had to be able to represent and distinguish between internal and external walls, doors, windows, openings, roof structure and miscellaneous structural elements. Linear elements such as columns and beams are represented as a line while planar areas such as walls and roof planes are represented as three dimensional planes bounded by straight lines. Each graphical element has an index to allow checking for duplicate elements to avoid over measuring.

Points are simply stored with an index and the three coordinates and lie in the positive x, y, z sector. The indices of the end points of lines are stored as well as the normal form of the three dimensional equation of the line, the length of the line and the indices of the planes for which the line is a boundary segment. Planes are stored as a three dimensional equation together with lists of the enclosing and enclosed boundary lines and the area of the plane. The equations of the lines and planes are needed for consistency checking and to allow detection of continuous lines and planes. Walls are stored as planes. When a door is inserted in a wall panel the list of perimeter lines is altered. The insertion of a window defines a hole in the plane, which is stored as a separate boundary.

Database Management

The use of Prolog as the programming language has significant advantages for a system that will grow to suit the user's needs. Prolog contains powerful database procedures which can be used to ensure that the user interface provides access to all of the data stored in the system, even after extensive modification by the user. For example, the dialogue that allows the user to select the construction types for the building components uses the "findall" statement to create a list of all of the construction types that have been defined. The list of choices will automatically change as users modify the data to suit their methods of work and types of buildings.

Expanding TES

The extension of TES to incorporate other simulation systems has been catered for by using Inter-Application Communication (IAC) to communicate between the various programs which make up TES. New functionality can be added by writing a separate program to perform the function and then adding the required IAC calls to TES's shell to provide the data and to accept the response. It would even be possible to replace sections of TES with other software. For example, AutoCAD could be modified to replace TES's graphical interface using the AutoCAD Development System (AutoDesk, 1992).

An advantage of the modular strategy used in TES is that all of the modules do not have to reside on the same computer since they are independent programs which rely on a computer network for communication. Consequently a degree of parallelism is built in to TES. Processing speed is no longer solely dependant on the speed of one computer since the processing load can be shared amongst several machines across a network.

In its current form TES can only handle single zone simulations. The use of frames to represent building information is expected to simplify the extension of TES to more complex systems. The modular nature of the components of TES means that the shell is the only module that will need to be altered to accommodate more complex simulations. The supply of information on the building elements separating zones would be a relatively simple matter. The adaptability of the code in the simulation programs would determine how readily a separate multi-zone simulation program could be integrated with the shell.

Conclusions

TES has demonstrated the integration of standard CAD interfaces with thermal simulation programs is possible and can significantly enhance the usefulness of thermal simulation. This is achieved by the use of knowledge based techniques and a blackboard architecture which have solved some of the problems inherent in the use of technical computer programs by untrained designers. The automation of data preparation from the CAD front end and the ability to give advice rather than tables of numbers are the major contributors to this ease of use. The use of Prolog and Flex rather than the traditional algorithmic languages has significant advantages for the ease of programming and user extensibility.

References

Autodesk (1992), *AutoCAD Development System Programmer's Reference*, AutoDesk

Ballinger, J.A., Prasad, D.K. & Rudder, D (1992) *Energy Efficient Australian Housing* , 2nd Ed, Australian Government Publishing Service

Brown, G.Z. & Sekiguchi T. (1989) Energy Scheming 1.0 *Proc 1989 Congress of the International Solar Energy Society* , Kobe City, Japan

Clarke, J.A. & Maver T.W. (1991) Advanced Design Tools for Energy Conscious Building Design: Development and Dissemination. *Building and Environment*, Vol 26 No. 1, pp 25-34

Clockson, W.F. & Mellish, C.S. (1984), *Programming in Prolog*, 2nd Ed, Springer Verlag

Englemore R. & Morgan, T (eds) (1988), Blackboard Systems, Addison-Wesley Pub. Co.

Ferry D.J. & Brandon, P.S. (1991) *Cost Planning of Buildings* , BSP Professional Books

Harrington-Lynn, J. (1974), The Admittance Procedure, Variable Ventilation. *Building Services Engineer* , 42, November

LPA. (1989), *Flex Expert System Toolkit*, 3rd Ed. Logic Programming Associates Ltd.

Maher, M.L. (ed) (1987) *Expert systems for Civil Engineers : Technology and Application.* American Society of Civil Engineers, New York

Markus, T.A. & Morris E.N. (1980) *Buildings, Climate and Energy* . Pitman Publishing

Minsky, M. (1985), A Framework for Representing Knowledge. In *Brachman, R.J. & Levesque, H.J. (eds) Readings in Knowledge Representation*, Morgan Kaufmann, pp245-262

Szokolay, S.V. (1990), *Thermal Design of Buildings* , RAIA Education Division

Szokolay, S.V. (1992), *ArchiPak User Manual, Version 4.2* , Architectural Science Unit, University of Queensland, Australia