

Performance Modelling and Conceptual Design

Andrew Marsh *

The University of Western Australia

Paper presented at International IBPSA Conference, Sydney, December 1996

A number of interactive design tools have recently been developed by the author. These tools differ from many similar packages in that they have been developed from a design perspective, to allow the performance analysis of buildings at any stage in the design process. In addition, within the one application, simultaneous analysis of thermal performance, daylight and electric lighting levels, shading and overshadowing patterns, acoustic response, transmitted sound levels, approximate overall cost and total environmental impact is possible. This means that the ramifications of any design decision can quickly be compared on many different levels without having to input the same data in different formats to a number of different tools.

This work has required a new approach to modelling and information gathering. The tools are all based on fully interactive 3D models. Whilst CAD data can be imported and the process of geometric input is no more complex than traditional CAD applications, the model itself has been imbued with a great deal more architectural and material information. This means that most of the detailed data required by any analysis engines can be automatically extracted from the model by simple geometric analysis. This paper introduces the conceptual background to this work and details the modelling system used and the data structures required.

Introduction

This work was initiated in response to the apparent low priority being given to building performance issues within the architectural profession, especially at the most conceptual stages of design. It was quickly found that this was not necessarily from a lack of interest on the part of architects, but from a general lack of tools.

Whilst there are many excellent architectural science software packages currently available, they appear to have a very low market penetration amongst architecture firms. At least in Perth, meaningful performance analysis only ever occurs at the very end of the design process, if at all. This means that poor results only ever initiate minor or cosmetic changes, when a major rethink is often required.

One of the reasons for this is the nature of most architectural science software. Given the complexity of the processes being modelled, an enormous amount of very detailed information is often required. It becomes the user's task to extract this information from the design and present it in a

format that each particular application can understand. There are currently no standards for these formats, so each application is substantially different.

This can be a tedious process, especially if it involves the creation of fixed format text files. More importantly, such processes often require some far-reaching assumptions and major simplifications of the actual model. Many applications also restrict the model to an abstract 2D representation.

All of these problems act to psychologically separate the process of analysis from the process of design. As a result, such tools are seldom used at the conceptual stages. However, that is exactly where they are required, to guide the aesthetic decision-making process toward a more efficient building.

Just Playing Around

At the conceptual stage most architects are simply playing around with basic ideas. This was considered to be of fundamental importance. One of the aims of much of the author's work has been to re-introduce the art of play into architectural science.

In order to play around, there needs to be something

andrew@arch.uwa.edu.au

The School of Architecture and Fine Arts
The University of Western Australia
NEDLANDS 6009

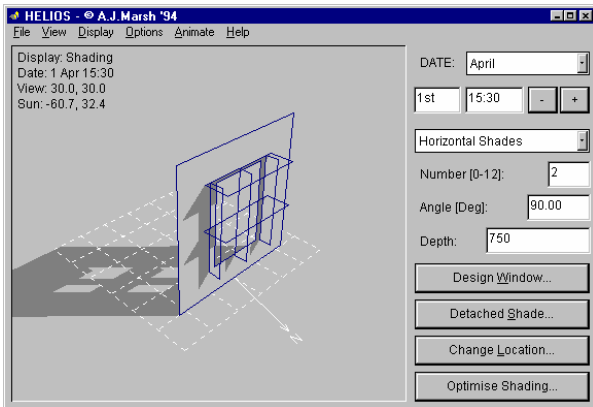


Figure 1: An interactive shading device design tool analysis.

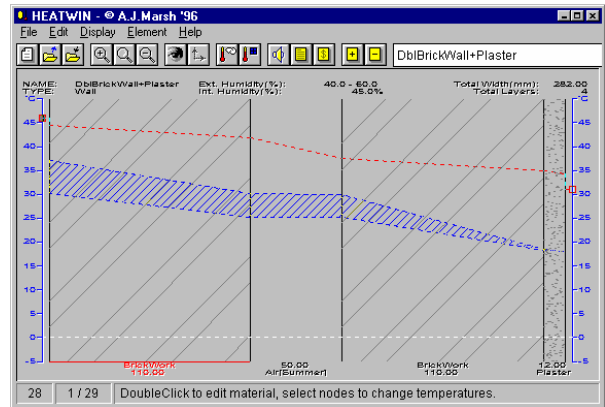


Figure 2: Material design with heatflow and condensation analysis.

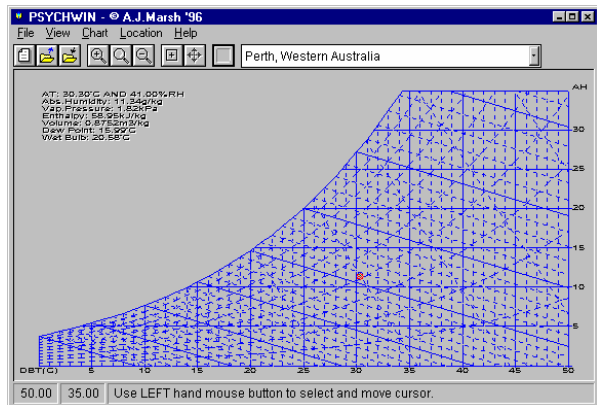


Figure 3: Interactive psychrometric and climatic analysis.

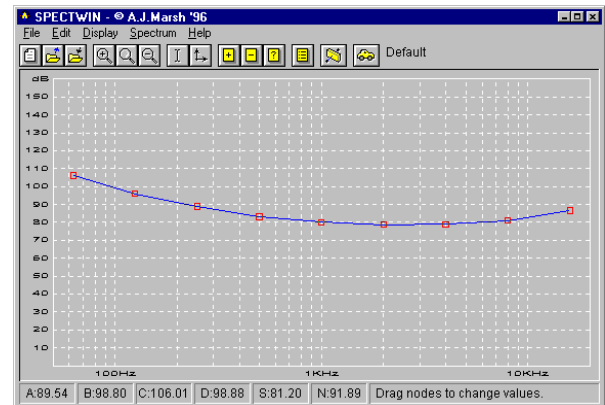


Figure 4: Acoustics, traffic noise prediction and barrier design.

These figures show the initialisation screens from some of the applications that form the suite of design tools. The intent is to provide something to begin playing with immediately on startup. Users can select nodes in each model and begin dragging to determine the effects. As ideas are resolved, more detailed parameters can be set in dialogue boxes and location selectors.

to play with. Many analysis tools require at least half an hour of data entry before even the most basic model is ready to test. It was therefore decided that any tools developed would fully initialise on startup, with at least something ready to go immediately. This idea formed the basis for a number of packages in the suite (Figures 1-4). It meant creating very basic parametric models or graphs with commonly occurring situations already displayed. From this initial stage, more detailed parameters can be adjusted as the model is resolved and more information becomes available.

Having introduced these applications into a number of architectural science courses, the response from students has been positive. In addition to instant-start ideas, observation of student behaviour makes it apparent that a fully interactive interface and immediate visual feedback are extremely important factors in encouraging further experimentation and 'design'. As a result, substantial work has been done in both of these areas.

An Interactive Interface

The aim of any interface is to maximise information input whilst minimising effort. Interfaces for the generation of 3D geometry have always been reasonably cumbersome due to the need for a working plane on which to situate the mouse

pointer. This can make the process of creating complex 3D geometry somewhat difficult by requiring it to be created in one plane, and then angled and rotated into place, or having the user constantly manipulating their working plane.

An Intuitive Cursor System

Given the intrinsic problem of situating the mouse pointer in 3D, an intuitive system has been developed that is based on the three axial planes. When moved normally, the mouse moves around the XY plane at Z=0. Hold down the SHIFT key and the mouse moves in the YZ plane, based at its current position in X. Move the mouse up in Z and, when the SHIFT key is released, the XY plane has been repositioned in Z. Similarly, the CONTROL

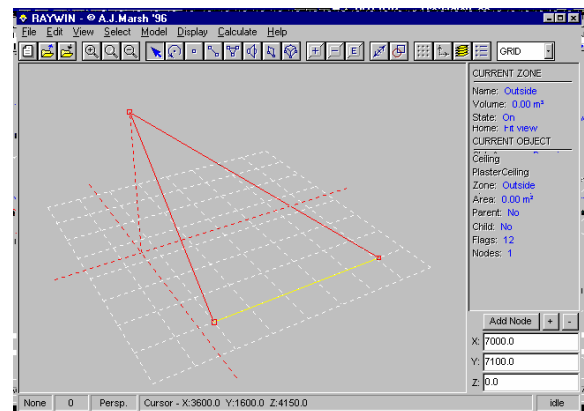


Figure 5: Shows the 3D mouse cursor at work creating an inclined plane.

key moves the mouse in the XZ plane.

When coupled with traditional object snap points, this system is very simple to use and quick to pick up. It allows the user to work interactively in 3D with complete freedom. A visual cursor consisting of two lines at $Z=0$ running in both the X and Y directions, and a line running from $Z=0$ to the height of the cursor, is displayed whenever the 3D mouse is active (Figure 5).

Spatial and Relational Modelling

Fundamental to the model used here is the concept of modelling space. Typical CAD applications use a set of simple primitives out of which complex geometric constructions can be generated. The existence of enclosed space is simply a by-product of defined surface geometry.

In this work, the model consists of a series of defined spaces. The application can test each space for adjacency with any other and then, by examining the area of interface, determine exactly how the two will interact. This means that the designer now works by arranging spaces, or *zones*, instead of a series of unrelated geometric primitives.

Obviously geometric primitives are needed in order to define and enclose each zone, however, they now exist as a by-product of the space, not the other way around. Interactively moving a zone moves all of its defining elements, whilst moving an element simply redefines the extents of the zone.

This approach facilitates interactive manipulation far more than traditional CAD applications. Whilst the definition of a zone as a complex closed polyhedra was considered far too restrictive, significant work has been done on the implicit relationships between architectural and structural elements to automatically generate zonal geometry. This has resulted in a system of relational mapping between specific elements that greatly simplifies user input and significantly enhances the editability of the model. Relationships are established during construction of the model and consist mainly of spatial links between nodes and a referential parent-child hierarchy.

This means that manipulating the shape of a floor, from which walls and a ceiling have been derived, will automatically update the position and shape of those derived elements. This includes rotating and resizing walls, relocating child windows and doors within moved walls and reshaping the ceiling to fit. Whilst this is a very simple example, the system

extends to any geometry.

A zone may be defined by any number of complex elements of any shape, and assigned any number of different materials. These defining elements do not need to be related in any way, or even fully enclose the zone. As a result, all relationships between elements are completely dynamic and easily modified or removed. Such an arrangement allows zones to contain multiple floors, which need not necessarily even be flat. They may also contain any number of internal partitions, suspended floors or ceiling panels.

Nodal Mapping

Most of the functional information about these implicit relationships is actually stored in the nodes that define each object. For example, nodes in an extruded wall are either locked to nodes in the floor from which it was derived or restricted to linear movement along the line of the extrusion vector. Similarly, each node in a child window has both an INSIDE and COPLANAR link to its parent plane.

Using list pointers, boolean flags and optimised analysis code means that storing and testing such relationships does not require much overhead at all. This means that all of the nodes in the child window can be tested whilst it is being interactively moved, making it possible, for example, to restrict movement to within the bounds of its parent.

Each node stores pointers to the relationships by which it defines its own position. These relationships are stored in the parent and child entities. As a result, altering one of these relationships automatically updates the position of any referencing nodes. Thus, changing an extrusion vector or moving a parent node does not require any protracted recalculation, simply an update of nodal positions carried out during a redraw.

Building Elements

The process of relationship mapping imbues the model with significant architectural information. Defining each object as a building element means that the application can apply default materials intelligently and knows what architectural role each object plays. Properties specific to these elements can then be used in performance calculations. Also, useful calculations such as total floor area, window area, distance to fire escapes, total installed flux and the like become quite trivial.

The basic building elements currently used are as follows: *Void, Roof, Floor, Ceiling, Wall, Partition, Window, Panel, Door, Speaker, Light and Point.*

Material Specification

Once defined as a building element, the application automatically assigns a default material from a customisable library. The defaults for each element type can be individually customised for each library and model file. At any time, the material assignment of any object or group of objects can be changed in the Object Properties dialogue box (Figure 6).

The properties of assigned materials are used in all subsequent performance calculations. As a result, each material requires an extensive set of properties defining its heat flow, sound transmission, sound reflection, visual and textural qualities.

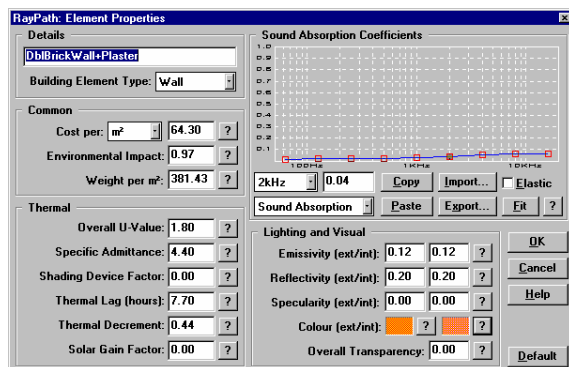


Figure 7: Element properties of an example double brick wall library material.

Figure 7 shows the material properties dialogue box. Much of this information is automatically calculated when the material is designed in the library manager (Figure 4). It is envisaged that manufacturers will provide their own frequently-updated libraries to practitioners should such a system gain widespread acceptance. The data is divided into groups as follows:

Thermal Properties

These include values for heat flow such as the U-value, specific admittance, thermal lag and decrement. Single figure values for solar gain factor and the effects of a shading device are also included. For windows, thermal lag and decrement become alternating solar gain factors for both light and heavy constructions. The emissivity of both internal and external is also stored, but as part of the visual and lighting group.

Visual and Lighting Qualities

As these are mostly surface properties, values are stored for both internal and external surfaces. These include emissivity, reflectivity, specularity and colour, which is stored as an RGB triple. Also the overall transparency of the element is stored as a fraction.

Acoustic Properties

Acoustic values are stored in two arrays holding sound absorption and transmission loss coefficients

for each octave band from 63Hz to 16kHz.

Miscellaneous

The weight and cost of each element is stored as a cost per unit or cost per metre squared, as appropriate to the individual element. In addition, a single figure index for environmental impact has been used. There is much developmental work being carried out in this area so this value will probably require significant updating.

Sources

Sources and receiver points can also be defined in the model. In addition to the miscellaneous properties of all elements, each source's power consumption, heat output and start/stop times are also stored. The directional properties are stored as a polar diagram, storing relative values over a full 180 degrees in 10 degree increments (Figure 8).

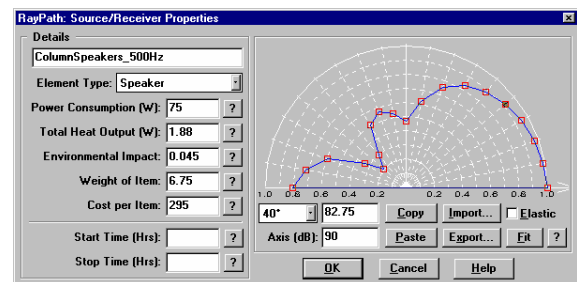


Figure 8: Source/receiver properties dialogue box.

Immediate Visual Feedback

Another important aspect of an interactive design tool was considered to be the visual quality of its feedback. As a result, all calculations are performed graphically on the model. Rays are displayed as they are generated, planes are highlighted as they are analysed and stereographic overshadowing is instantly updated whenever objects are moved. This allows a user to pick up information visually that may be difficult to detect mathematically.

Significant ongoing consideration is also being given to the presentation of calculated results.

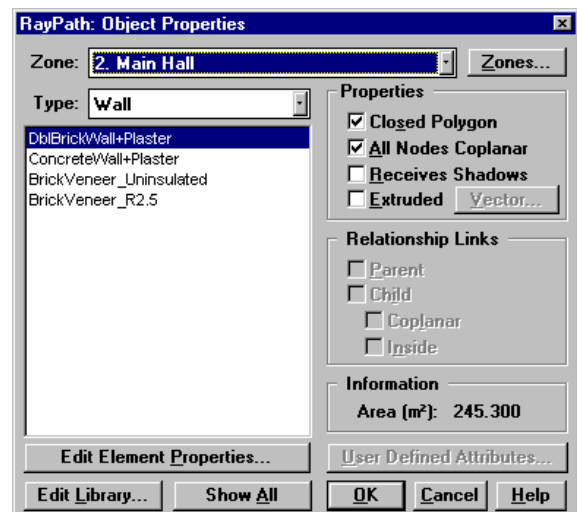


Figure 6: Object properties dialogue box detailing material assignment.

Much of this information is currently displayed within the model or on integrated graphs, however, it is difficult using this method to adequately compare before and after effects. As a result, a system of dialogue boxes for direct comparison is being developed.

Conclusion

It is hoped that this work will represent a small step forward in the development of interactive design-oriented interfaces for performance analysis tools. From the authors work, it is clear that integrated systems that compare performance in many areas are more attractive to designers than focussed tools. Whilst there is room for both, it is important to develop open tools that can be used to introduce architectural science issues much earlier in the design process than currently possible.

As architects and building designers begin to transact and share information electronically, it is important that there be some standards for the display of performance models as well as the representation of material properties and climatic information. From this, it follows that standard measures for the comparison of building performance and indexes that include a range of performance issues, including operational costs and environmental impact are just as important.

Acknowledgments

The author would like to thank Dr Derek Carruthers for his enormous patience and support in the supervision of this work.